2011 IEEE Intelligent Vehicles Symposium (IV)
Baden-Baden, Germany, June 5-9, 2011

# Velodyne SLAM

**Frank Moosmann** and **Christoph Stiller**
Department of Measurement and Control
Karlsruhe Institute of Technology, 76128 Karlsruhe, Germany
Email: frank.moosmann@kit.edu

***Abstract*—Estimating a vehicles' own trajectory and generating precise maps of the environment are both important tasks for intelligent vehicles. Especially for the second task laser scanners are the sensor of choice as they provide precise range measurements.**

**This work proposes an approach for simultaneous localization and mapping (SLAM) specifically designed for the Velodyne HDL-64E laser scanner which exhibits characteristics not present in most other systems. This comprises the continuous, spinning data acquisition and the relative high sensor noise. Together, these make standard SLAM approaches generate noisy maps and inaccurate trajectories. We show that it is possible to generate precise maps and localize therein in spite of not using wheel speed sensors or other information. The presented approach is evaluated on a novel, challenging 3D data set being made publicly available.**

## I. INTRODUCTION

Among the most basic tasks of an intelligent vehicle is estimating its own trajectory as it is a prerequisite for path-planning and control. Two kind of approaches can be distinguished, namely incremental and global ones.

Incremental approaches use a limited history of local sensor data and usually employ some recursive state filtering thereon. Used sensors include wheel speed sensors, inertial sensors, cameras, and laser scanners. Although astonishing results were reported for various types of sensors, the main disadvantage of incremental methods remains the accumulation of error over time. This disadvantage can only be overcome by using global information, which characterizes the second kind of approaches. Global information can be given by e.g. a GPS receiver or a global map. While GPS is inaccurate and not available in all environments, generating a detailed global map in advance is very costly. Fusing GPS into local approaches can solve the problem but imposes some restrictions on the application.

This motivated the development of SLAM methods for simultaneous localization and mapping as a semi-global approach. Thereby, the vehicle incrementally builds a map while it moves and localizes itself within that map. In contrast to incremental approaches, this results in less incremental error as the history of sensor data is potentially infinite. In addition, it allows place-recognition to close a loop and eliminate accumulated errors. Since SLAM benefits from dense data, the choice of sensor practically limits to cameras and laser scanners. This work concentrates on the latter, as typically both the precision and the field of view is larger.

A very good introduction to SLAM can be found in [1], which also includes the trends of the last years towards probabilistic techniques (*e.g.* Extended Kalman Filters [2], Unscented Kalman Filters [3], Sparse Extended Information Filters [4], or Rao-blackwellized Particle Filters [5]). Excellent results were obtained, but unfortunately, most of these methods are computationally hard when the number of landmarks[1] grows high. Not surprisingly, their application concentrated on the use of 2D laser scanners especially within buildings. Yet, Holz *et al.* [6] showed just recently that the combination of several heuristics with simple scan-matching yields a fast and nevertheless as accurate approach.

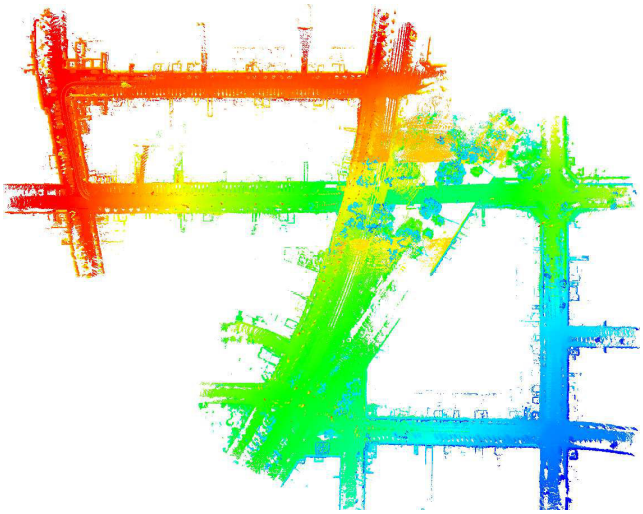


Fig. 1. Generated map for scenario 2 from a bird's-eye view perspective. Color encodes altitude.

When focusing on outdoor-environments, the use of full 3D data becomes inevitable. To keep device costs low, all current systems scan the 3D volume by turning a sensor in some way. Nüchter *et al.* [7] stop the vehicle for each scan and do incremental scan-matching. As this is not appropriate for intelligent vehicles, one has to cope for the sensor movement as scanning times cannot be neglected. One such approach was presented in [8] where a nodding SICK 2D-laser scanner was used. Another approach [9] spins a similar scanner around its center axis and estimates the trajectory during scanning within the scan-matching.

In this work, the Velodyne HDL-64E S2 laser scanner is used. Mounted on top of a vehicle, it continuously scans the complete 360° surroundings. This field of view combined with a high data rate made it popular among the intelligent vehicles

[1] In most cases raw measurements are used as landmarks

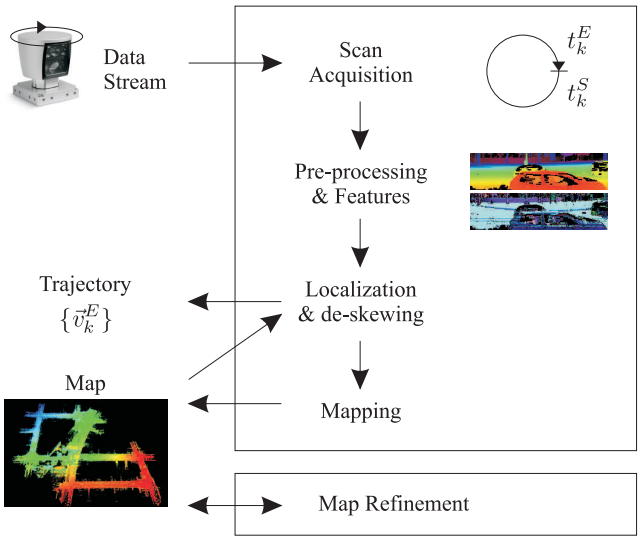



Fig. 2. Sketch of the proposed method

community, especially within the DARPA Urban Challenge. Although there exist approaches for re-calibration [10], [11], the main drawback is the relatively high measurement noise.

The approach presented here introduces a special handling of this noise within the context of SLAM. On building the map, measurements are adapted in areas of flat surfaces. This not only makes localization more precise (see also [12] within the context of TOF-cameras), it also enables the built maps to be used later on as 3D models. To get even more precise models, an off-line map-refinement step is introduced. In contrast to [8] no assumptions about the environment are made in this work. We also show that a simpler technique as opposed to [9] can be used to cope for the sensor movement.

This paper is organized as follows. The next section details the hardware set-up and the proposed algorithm. In Section III experimental validation is provided in two difficult multilayer outdoor scenarios. Section IV concludes this paper and gives an outlook to future research.

## II. PROPOSED METHOD

An overview of the proposed method can be seen in Fig. 2. Input is the continuous data stream of the Velodyne scanner, output is a detailed 3D map i.e. a point cloud of the environment along with the vehicle trajectory. The algorithm works in different steps, all of them detailed in the following.

### A. Scan Acquisition

A bit simplified, the Velodyne HDL-64E (S1/S2) contains a column of 64 laser diodes, covering a pitch range of approximately 26 degrees. While turning clockwise around the scanners' center axis, these laser beams continuously sweep the surroundings producing distance readings in fixed time intervals. The corresponding sweep angle is determined by a built-in rotation encoder. As the pitch angles of the lasers are fixed, each of the lower lasers produces a ring of point measurements on the road surface (if not occluded).

To enable the use of popular scan matching algorithms, the input stream is divided into chunks of 360° turns of the sensor[2]. In the following, these chunks are also called *scans*. We denote the scan index $k$ by subscript and the relative scanning angle by superscript, $S$ denoting the starting angle of a turn and $E$ the end angle. Scan $k$ is thus recorded between the two timestamps $t_k^S$ and $t_k^E$. Due to the continuous scanning characteristic $t_k^E = t_{k+1}^S$.

For further processing, the range measurements of one scan are arranged within a *range image* of 870x64 pixel, where each row corresponds to the measurements of one specific laser. The imprecision caused by this projection is low as compared to the sensor noise and will be compensated in Section II-E.

### B. Pre-processing

This section explains preprocessing and feature calculation given a two-dimensional array of range measurements $R : (u, v) \mapsto r$ as obtained in Section II-A (see also Fig. 3). To increase readability, we subscript functions by index instead of using pixel coordinates as function arguments: $R(u, v)$ is thus denoted by $R_i$. Connections are implicitly established from each pixel to its four neighbours, also denoted by indices:

$$i_1 := (u+1, v) \quad i_3 := (u-1, v) \quad i_5 := i_1$$
$$i_2 := (u, v-1) \quad i_4 := (u, v+1) \quad (i_1)_1 := ((u+1)+1, v)$$

Altogether, the following functions/images are used:

| | | | |
|---|---|---|---|
| range measurement | $R_i$ | : | $i \mapsto r$ |
| point coordinates | $\vec{P}_i$ | : | $i \mapsto (x, y, z)^T$ |
| distance vector | $\vec{D}_{i,j}$ | = | $\vec{P}_j - \vec{P}_i$ |
| linkage value | $L_{i,j}$ | : | $i, j \mapsto l$ |
| normal vector | $\vec{N}_i$ | : | $i \mapsto (n_x, n_y, n_z)^T$ |
| normal confidence | $C_i$ | : | $i \mapsto c$ |

Some of these relations are illustrated in Fig. 3 and their calculations are explained in the following.

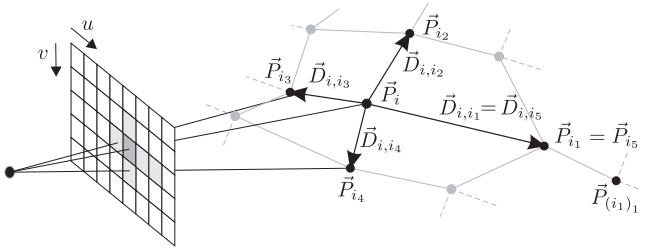



Fig. 3. Range image as implicit graph on 3D coordinates

The point coordinates (relative to the laser scanner) are directly obtained from the range measurements using the physical sensor set-up. The distance vectors follow immediately. The linkage measure is a first indication for grouping pixels together and it is used to weight calculations on pixel connections. A pixel connection gets assigned a high linkage value if neighbouring distance vectors have similar length.

[2]Note that this does not correspond to an exact 360° sweep of the environment when the vehicle is turning.

Exemplary, the linkage of a pixel to its right neighbour is calculated as:

$$L_{i,i_1} = \min(\text{sigm}(|\frac{(R_i - R_{i_1}) - (R_{i_3} - R_i)}{(R_{i_3} - R_i)}|), \\ \text{sigm}(|\frac{(R_i - R_{i_1}) - (R_{i_1} - R_{(i_1)_1})}{(R_{i_1} - R_{(i_1)_1})}|)) \quad (1)$$

The following sigmoid-like function serves as soft threshold:

$$\text{sigm}(x) = 0.5 - \frac{0.5(x - \theta_1)\theta_2}{\sqrt{1 + (x - \theta_1)^2 \theta_2^2}} \quad (2)$$

where $\theta_1$ specifies the effective threshold and $\theta_2$ is a constant scale parameter to influence the tangent inclination at the threshold.

Second, a local surface plain represented by its normal vector is estimated at each measurement. For a given pixel with its four neighbours the normal vector is calculated as the average of the four cross products, each weighted by the product of their linkage values:

$$\vec{N}'_i = \sum_{j=1}^{4} L_{i,i_j} L_{i,i_{j+1}} (\vec{D}_{i,i_j} \times \vec{D}_{i,i_{j+1}}) \quad (3)$$

A moving average filter is then applied to the field of surface normals in order to reduce noise:

$$\vec{N}_i = \frac{\sum_{j=1}^{4} \vec{N}'_{i_j}}{||\sum_{j=1}^{4} \vec{N}'_{i_j}||} \quad (4)$$

Third, a confidence value is estimated for each normal vector. For the horizontal and vertical direction it is separately evaluated how plain the neighbourhood is. For a given connection from $i$ to $j$, the angle of the distance vector to the plain defines a probability that the plain assumption holds for this connection:

$$C_{i,j} = \exp\{-\theta_3 \arcsin \left| \frac{\vec{D}_{i,j} \cdot \vec{N}_i}{||\vec{D}_{i,j}||} \right|^2 \} \quad (5)$$

The decay of the probability is controlled by parameter $\theta_3$. In a fuzzy-logical manner, the plain assumptions holds if it either holds horizontally or vertically. However, it is limited by the maximum linkage product from the normal calculation:

$$C'_i = \min(L_i^{\max}, \max(C_{i,i_1} C_{i,i_3}\ ,\ C_{i,i_2} C_{i,i_4})) \quad (6)$$

$$L_i^{\max} = \max{}_{j=1}^{4} L_{i,i_j} L_{i,i_{j+1}} \quad (7)$$

Median filtering is afterwards applied on the 4-neighbourhood to smooth the confidence values. From visual inspection, this method seems to produces better estimates than calculating a flatness-criteria from the eigenvalues of the principal component analysis.

### C. Localization

Once the surfaces of the current scan $\{S_i = (\vec{P}_i, \vec{N}_i, C_i)\}$ are known, localization takes place. Hence, the goal is to estimate the pose of the vehicle[3] $\vec{v}(t)$ relative to some global coordinate frame. $t$ thereby varies between the scanning start time $t_k^S$ and scanning end time $t_k^E$. We assume motion with constant velocity which lets us linearly interpolate between $\vec{v}(t_k^S)$ and $\vec{v}(t_k^E)$, also denoted $\vec{v}_k^S$ and $\vec{v}_k^E$ respectively. Since $\vec{v}_k^S = \vec{v}_{k-1}^E$, it is for a given scan $k$ sufficient to estimate $\vec{v}_k^E$, the vehicle pose at the end of the scan.

To estimate $\vec{v}_k^E$, the current scan is matched against the whole map, similar to [6]. Because the map is built from measurements, it is in this work just a collection of surfaces in a world reference frame $\{s_i = (\vec{p}_i, \vec{n}_i, c_i)\}$, *i.e.* point coordinates along with normal vectors and normal confidence values. The current vehicle position is estimated by searching a pose that results in a *best fit* (thus $\vec{v}_k^E = \arg\min_v E(v)$) according to the energy

$$E(v) = \sum_{i \in \text{scan}} (\vec{n}_{\text{NN}(i)}^T (\text{T}(\vec{P}_i, v) - \vec{p}_{\text{NN}(i)}))^2 \quad (8)$$

This is the sum of squared errors of the scan points $\vec{P}_i$ transformed by the vehicle pose $v$ into the global coordinate frame and projected onto the surface of the nearest neighbour in the map. This miminization is solved using the popular *Iterative Closest Points* algorithm (ICP) as described in [13]. The algorithm implements iterative minimization which we initialize with the last pose predicted by the last movement

$$\tilde{\vec{v}}_k^E = \vec{v}_{k-1}^E \oplus (\vec{v}_{k-1}^E \ominus \vec{v}_{k-1}^S) \quad (9)$$

At each iteration, the current pose is used to search the nearest neighbours in the map. The pose is then updated by minimizing the energy. The nearest-neighbour-search is thereby carried out in 6D $(p_x, p_y, p_z, n_x, n_y, n_z)$. This makes the convergence more robust against inital transformation errors. To speed up the algorithm only a subset of the current scans' surfaces is used. 1000 surfaces are sampled uniformly from the upper half of the image and 500 surfaces from the lower half. More sophisticated samplings ([14], [15]) seem not to be necessary.

### D. De-skewing

The core of the ICP algorithm transforms at each iteration the current scans' surfaces into world coordinates using the current estimate of the vehicle pose. This does not yet account for possible sensor movements during scan acquisition of the Velodyne scanner.

One possible solution is to move the estimate of the whole trajectory into the ICP core, as in [9]. This, however, sophisticates the algorithm and makes need for non-linear solvers. As the Velodye scanner operates at a relatively high scanning rate of $\sim 10$Hz a simple but effective technique we call *de-skewing* is applied:

Initially, the last pose $\vec{v}_{k-1}^E$ and the predicted vehicle pose $\tilde{\vec{v}}_k^E$ are used to linearly transform the surfaces of the current scan relative to $\tilde{\vec{v}}_k^E$. Then the ICP is run on the de-skewed image. Finally, the image is de-skewed once more, but using the final pose $\vec{v}_k^E$. Theoretically, this de-skewing could be employed at each iteration of the ICP, but experiments showed that it is sufficient to de-skew only twice: before and afterwards.

[3]The vehicle pose is defined as the position and orientation of the laser scanner. For the first frame this can be arbitrarily initialized

### E. Mapping

As already mentioned in Section II-C, a map is in this work a loose collection of surfaces in a world reference frame $\{s_i = (\vec{p}_i, \vec{n}_i, c_i)\}$. The map is stored in a 3D grid structure with each cell containing maximally one surface according to its 3D position $\vec{p}_i$. The grid resolution $g$ hence defines the level of detail of the map.

Once the current scan is localized within the map and de-skewed[4], the map is up-dated in two steps.

First, each measured surface $S_i$ is adapted according to its neighbours in the map. The idea in regions of high normal confidence is to move the 3D point coordinate $\vec{P}_i$ along the normal vector $\vec{N}_i$ to

$$\vec{P}_i'(a) = \vec{P}_i + a \cdot \vec{N}_i \tag{10}$$

until it best represents a plain together with the neighbouring surfaces. The weighted point-to-plain-energy is defined similar to the ICP-energy by

$$\begin{aligned} E_{S_i}(a) &= \sum_{j \in \text{kNN}(S_i)} w_{ij} (\vec{n}_j^T (\vec{P}_i'(a) - \vec{p}_j))^2 \\ w_{ij} &= C_i c_j \vec{N}_i^T \vec{n}_j \end{aligned} \tag{11}$$

where kNN returns $k$ map-surfaces in a specified neighbourhood and $w_{ij}$ are weights according to normal confidence and similar normal direction. The adjustment $a$ is determined by $\hat{a}_i = \arg\min_a E_{S_i}(a)$, which is a closed-form least-squares solution.

This adaptation is the key step to account for the measurement noise of the Velodyne scanner and for imprecisions in the localization and de-skewing steps. As it avoids flat areas to grow perpendicular to the plain, further localization is improved since the ICP energy function then has a well-defined minimum.

Second, each measurement is added to the map. If the corresponding cell is non-empty, its surface $s_i$ is replaced by the current measurement $S_i$ in case

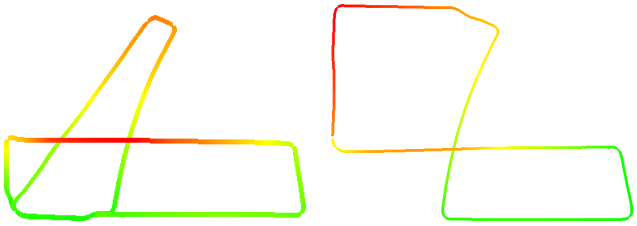
$$\frac{r_i - R_i}{r_i} + (C_i - c_i) > \theta_4 \tag{12}$$

holds. Hence, preference lies on surfaces that have a higher normal confidence and/or points that were captured from lower distance. As a consequence, cells of the map are never erased and the map grows continuously. To limit the computational load, it is possible to delete or store cells to disk if they move out of a certain range of the vehicle.

### F. Map Refinement

Most SLAM methods build the map just for localization purposes. City maps, on the contrary, are mostly built using high-precision laser scanners which need several seconds or even minutes for one 360° scan. In the following it is shown that the accumulated map can be even further refined to obtain a final map containing more details.

[4]The first scan is belatedly de-skewed and the map readjusted once the second scan was localized



(a) Scenario 1, length=1.3km (b) Scenario 2, length=1.1km

Fig. 4. Sketch of the scenes used for evaluation. Color encodes altitude (green=low, red=high). In both scenarios, the horizontal crossing in the middle is a bridge.

The idea for refinement is similar to the adaptation step in Section II-E. In areas of high confidence of the normal vectors, measurements can be constricted onto a local plain. In Section II-E incoming measurements are adapted according to already existing, adapted neighbours. This is suboptimal as it first allows for increasing adaptation drift and second will not include future measurements.

To overcome these disadvantages, the map refinement builds a complete new map using the old map. In order to do so, it is here assumed that for each surface $\{s_i = (\vec{p}_i, \vec{n}_i, c_i)\}$ also the original, non-adapted measurement location $\vec{o}_i$ is stored. For each surface in the existing map $k$ nearest neighbors (kNN) are searched in a specified neighbourhood and, as in Section II-E, the energy

$$\begin{aligned} E_{s_i}(a) &= \sum_{j \in \text{kNN}(s_i)} w_{ij} (\vec{n}_j^T (\vec{p}_i'(a) - \vec{o}_j))^2 \\ w_{ij} &= c_i c_j \vec{n}_i^T \vec{n}_j \end{aligned} \tag{13}$$

is minimized for $a$ in order to obtain the surface location $\vec{p}_i'(a) = \vec{o}_i + a \cdot \vec{n}_i$ in the new map.

## III. EXPERIMENTS

The proposed algorithm was evaluated on two complex outdoor scenes captured with our experimental vehicle *AnnieWay*. The data is publicly available on http://www.mrt.kit.edu/z/publ/download/velodyneslam/. Sketches of the two trajectories are depicted in Fig. 4, which are both results of the proposed method. It is clearly visible that there is no jump in the position estimation and that loops close nicely which shows the quality of the proposed method.

In a first step, the estimated trajectory was compared against the recordings of an integrated navigation system (INS) which fuses GPS, wheel speed sensors and inertial measurements. However, local errors of the INS are much higher than those of the proposed method. Even after driving more than one kilometer, errors of the proposed method are in the same range (see Table I). This prohibits to use the INS as ground-truth.

In order to quantitatively evaluate the algorithm nevertheless, we used the characteristics of the scenarios to generate ground-truth for the end-position of the trajectory: As both scenarios are loops, we used ICP to match the last scan against the first scan to obtain the desired end-position. We thus

TABLE I
INFLUENCE OF THE ALGORITHM STAGES ON THE END-POINT ERROR

| Setting | Mapping | De-skewing | Adaptation | Error |
|---|---|---|---|---|
| Scenario 1 | | | | |
| IMU | – | – | – | 3.30 m |
| 1 | no | no | no | 19.60 m |
| 2 | no | no | yes | 19.33 m |
| 3 | no | yes | no | 27.41 m |
| 4 | no | yes | yes | 27.21 m |
| 5 | yes | no | no | 4.47 m |
| 6 | yes | no | yes | 4.13 m |
| 7 | yes | yes | no | 2.90 m |
| 8 | yes | yes | yes | 2.29 m |
| Scenario 2 | | | | |
| IMU | – | – | – | 2.64 m |
| 1 | no | no | no | 22.25 m |
| 2 | no | no | yes | 22.09 m |
| 3 | no | yes | no | 18.58 m |
| 4 | no | yes | yes | 18.89 m |
| 5 | yes | no | no | 8.08 m |
| 6 | yes | no | yes | 7.36 m |
| 7 | yes | yes | no | 4.81 m |
| 8 | yes | yes | yes | 4.10 m |

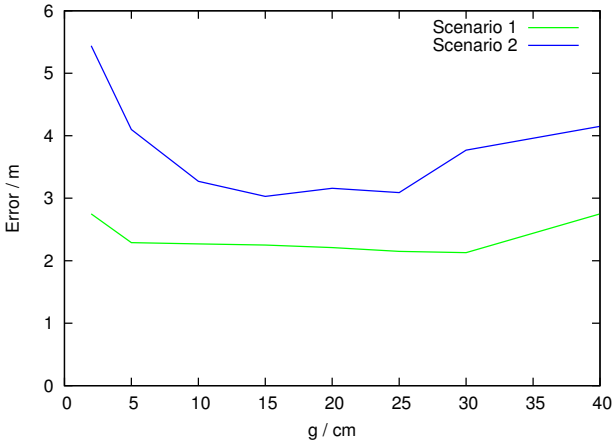


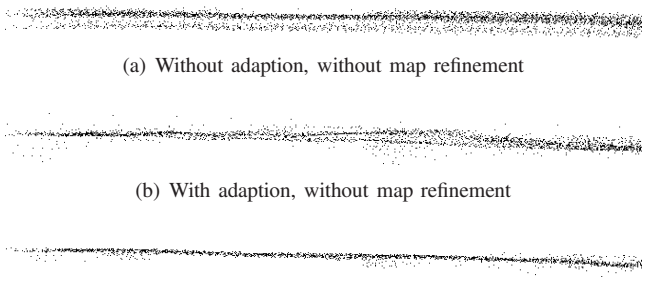

Fig. 5. Influence of the grid resolution $g$ on the end-point error.

evaluate the algorithm by defining the error as the euclidean distance between the estimated and the desired end-position.

The main focus in the evaluation is the long-term drift in the position estimation. Therefore, the implementation was altered in a way that parts of the map are discarded as soon as they get out of the viewing-range of the sensor (limited to 50m in the experiments). Although the used scenarios contain loops, no loop-closure was carried out when revisiting the same places. Hence, the calculated error is equally representative for scenarios without loops. If not otherwise specified, the parameters were fixed to $\theta_1 = 4$, $\theta_2 = 1$, $\theta_3 = 0.5$, $\theta_4 = 0.3$, and $g = 5\,\text{cm}$ throughout all experiments.

The algorithms' capability for localization was evaluated by (de)activating different stages. The results are listed in Table I. Setting 5 thereby corresponds to approaches like [7] and setting 7 is comparable to [9]. Not surprisingly, using pairwise scan-matching only (*i.e.* no mapping) results in very high errors, whereas with activated mapping localization is much more precise. According to Fig. 5, the grid resolution $g$ can thereby be chosen in a wide range – only the detail of the map suffers. Not accounting for the sensor rotation is the second principal influence of localization error. The linear interpolation used in this work thereby shows its effectiveness. Finally, adapting measurements while adding them to the map again improves the results.

Another focus of the evaluation is the quality of the produced map. The more detailed the map is, the better it can be used for other purposes, as *e.g.* city modelling. As this is hard to evaluate quantitatively, we here stick to visual inspection. An example map is shown in Fig. 1, one part of that map is enlarged in Fig. 7, an even smaller part is depicted in Fig. 6. It is clearly visible that both adaptation during mapping as well as the map refinement step helps in getting detailed surfaces.



(a) Without adaption, without map refinement

(b) With adaption, without map refinement

(c) With adaption, with map refinement

Fig. 6. Local point cloud of a road surface

The results show that the algorithm can be used for both, precise localization and city-model building. The grid size $g$ is thereby the main parameter to choose. A low value allows for high-precise maps, a higher value for faster (online) processing.

## IV. CONCLUSIONS AND FUTURE WORK

This work proposed an approach for simultaneous localization and mapping (SLAM), specifically designed for the popular Velodyne laser scanner. A new dataset was created and made publicly available to ease future evaluation. Experiments revealed that the proposed approach is more precise in estimating the trajectory than an integrated navigation system. A filtering step was proposed to refine the produced map offline in order to be used as detailed city-map.

Although the results of the algorithm are very convincing, there is still room for improvement. One of the next steps could be to detect and handle loop-closure appropriately as *e.g.* in GraphSLAM. Other future work will include to combine this approach with obstacle detection and tracking as the approach is currently limited to (nearly) static scenes.

## V. ACKNOWLEDGMENTS

The authors gratefully acknowledge the contribution of the German collaborative research center on Cognitive Automobiles (SFB/Tr28) granted by Deutsche Forschungsgemeinschaft.

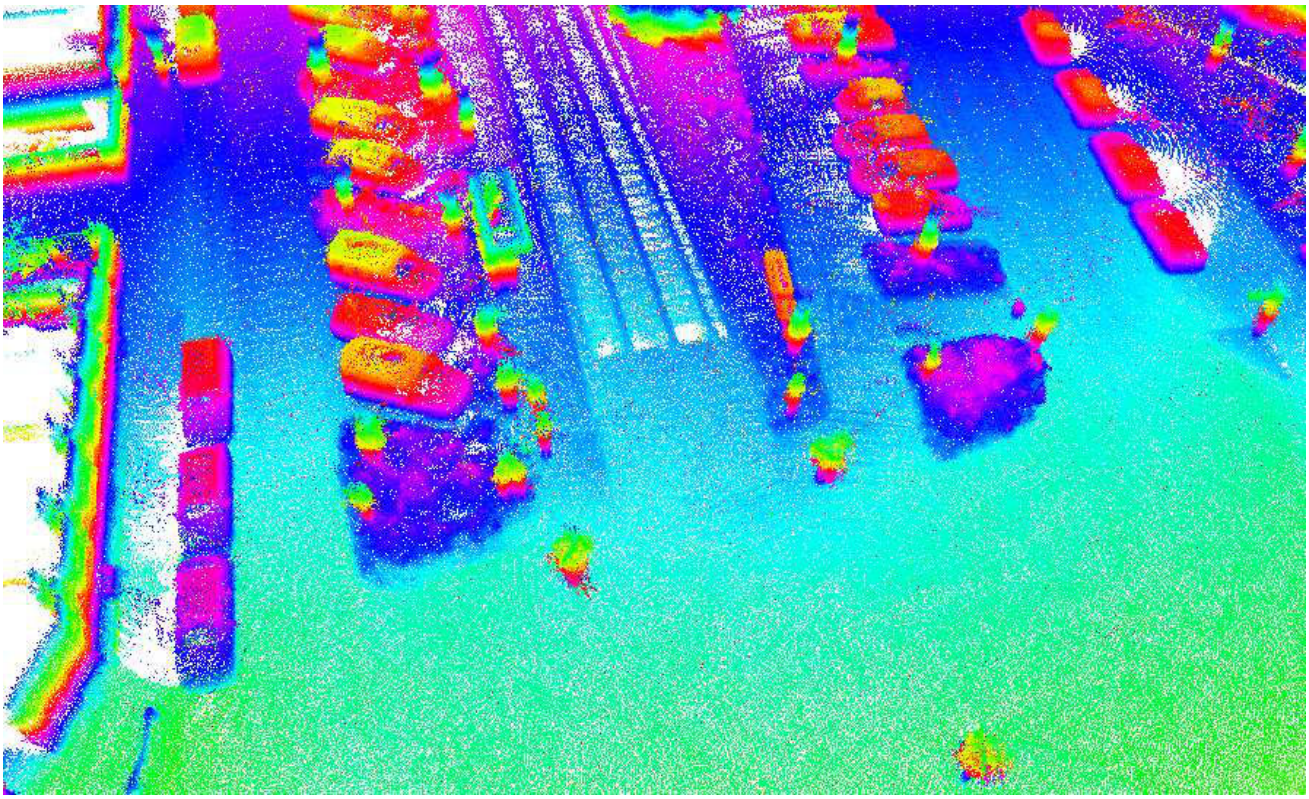


Fig. 7. Enlarged part of the map of scenario 1: Crossing of railway tracks. Color encodes altitude.

## REFERENCES


[1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*, ser. Intelligent robotics and autonomous agents. MIT Press, September 2005.

[2] J. J. Leonard, H. Jacob, and S. Feder, "A computationally efficient method for large-scale concurrent mapping and localization," in *Proceedings of the Ninth International Symposium on Robotics Research*. Springer-Verlag, 1999, pp. 169–176.

[3] D. Chekhlov, M. Pupilli, W. Mayol-Cuevas, and A. Calway, "Real-time and robust monocular slam using predictive multi-resolution descriptors," in *2nd International Symposium on Visual Computing*, November 2006.

[4] S. Thrun, Y. Liu, D. Koller, A. Y. Ng, Z. Ghahramani, and H. Durrant-Whyte, "Simultaneous Localization and Mapping with Sparse Extended Information Filters," *The International Journal of Robotics Research*, vol. 23, no. 7-8, pp. 693–716, 2004.

[5] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34 –46, February 2007.

[6] D. Holz and S. Behnke, "Sancta Simplicitas – On the efficiency and achievable results of SLAM using ICP-Based Incremental Registration," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, Alaska, USA, May 2010, pp. 1380–1387.

[7] A. Nüchter, K. Lingemann, J. Hertzberg, and H. Surmann, "6D SLAM—3D mapping outdoor environments: Research articles," *Journal of Field Robotics*, vol. 24, no. 8-9, pp. 699–722, 2007.

[8] A. Harrison and P. Newman, "High quality 3D laser ranging under general vehicle motion," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2008, pp. 7 – 12.

[9] M. Bosse and R. Zlot, "Continuous 3D scan-matching with a spinning 2D laser," in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE. Kobe, Japan: IEEE Robotics and Automation Society, May 2009, pp. 4312–4319.

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

[11] C. Glennie and D. D. Lichti, "Static calibration and analysis of the velodyne HDL-64E S2 for high accuracy mobile scanning," *Remote Sensing*, vol. 2, no. 6, pp. 1610–1624, 2010.

[12] S. May, S. Fuchs, D. Droeschel, D. Holz, and A. Nüchter, "Robust 3D-Mapping with Time-of-Flight Cameras," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, St. Louis, Missouri, USA, October 2009, pp. 1673–1678.

[13] Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 3, 9-11 April 1991, pp. 2724 – 2729.

[14] N. Gelfand, L. Ikemoto, S. Rusinkiewicz, and M. Levoy, "Geometrically stable sampling for the ICP algorithm," in *International Conference on 3-D Digital Imaging and Modeling*, 2003, pp. 260–267.

[15] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *International Conference on 3-D Digital Imaging and Modeling*, 2001, pp. 145–152.