VEHICLE LOCALIZATION USING LANDMARKS OBTAINED BY A LIDAR MOBILE MAPPING SYSTEM

Claus Brenner

Institute of Cartography and Geoinformatics
Leibniz Universität Hannover
Appelstrasse 9a, 30167 Hannover, Germany
claus.brenner@ikg.uni-hannover.de

Commission III, WG III/1

KEY WORDS: Mobile Laser Scanning, Feature Extraction, Mapping, Localization, Accuracy

ABSTRACT:

Accurate and reliable localization in extensive outdoor environments will be a key ability of future driver assistance systems and autonomously driving vehicles. Relative localization, using sensors and a pre-mapped environment, will play a crucial role for such systems, because standard global navigation satellite system (GNSS) solutions will not be able to provide the required reliability. However, it is obvious that the environment maps will have to be quite detailed, making it a must to produce them fully automatically. In this paper, a relative localization approach is evaluated for an environment of substantial extent. The pre-mapped environment is obtained using a LIDAR mobile mapping van. From the raw data, landmarks are extracted fully automatically and inserted into a landmark map. Then, in a second campaign, a robotic vehicle is used to traverse the same scene. Landmarks are extracted from the sensor data of this vehicle as well. Using associated landmark pairs and an estimation approach, the positions of the robotic vehicle are obtained. The number of matches and the matching errors are analyzed, and it is shown that localization based on landmarks outperforms the vehicle's standard GNSS solution.

1 INTRODUCTION

Mobile mapping systems using laser scanners combined with GNSS and inertial sensors are well suited for the production of large scale maps, since they reach a relative accuracy of down to a few centimeters. With a rate of more than 100,000 points per second, they capture the environment in great detail (Kukko et al., 2007). The problem of building a representation of the environment using laser scanners has also been investigated in robotics. Iconic representations, such as occupancy grids, have been used as well as symbolic representations consisting of line maps or landmark based maps (Burgard and Hebert, 2008). One of the major problems to solve is the simultaneous localization and mapping (SLAM), which incrementally builds a map while a robot drives (and measures) in unknown areas (Thrun et al., 2005). Often, 2D laser scans (parallel to the ground) are used for this, but it has also been extended to the 3D case (Borrmann et al., 2008).

Similar developments take place in driver assistance research. Active safety systems should completely prevent accidents, instead of just reducing the consequences. Therefore, vehicles are using more and more on-board sensors such as cameras, laser scanners or radar to gather spatial information about their environment. Compared to standard car navigation systems, these so-called advanced driver assistance systems (ADAS) operate at a very detailed scale, requiring large scale maps. The requirements for future maps and the possible capturing methods have been investigated in the NextMAP project (Pandazis, 2002). Some investigations with regard to the required detail and accuracy have been made in the 'Enhanced Digital Mapping' project (EDMap, 2004). ADAS applications have been classified, according to their requirements, into 'WhatRoad','WhichLane', and 'WhereInLane', the latter requiring a mapping accuracy of ± 0.2 m. It was concluded that contemporary mapping techniques would be too expensive to provide such maps for the 'WhichLane' and 'WhereInLane' case.

A key observation is that these findings assume a 'traditional' map production, i.e. the map is a highly abstract representation in terms of a vector description of the geometry, with additional attributes. However, depending on the application, this is not always necessary. For example, the accurate positioning of vehicles using relative measurements to existing map features does not necessarily require a vector map in the usual sense. Rather, any map representation is acceptable as long as it serves the purpose of accurate positioning. This means that map production to derive a highly abstract representation (which usually involves manual interaction) is not required, but rather a more low-level representation is suitable as well (which can be derived fully automatically and thus, inexpensively).

This approach is investigated in this paper. Since it would be unreasonable to provide dense georeferenced point clouds for the entire road network, a landmark based map is used instead, which has very low storage requirements. Poles are extracted fully automatically from an extensive 22 km Lidar mobile mapping campaign and stored in a reference data base. Then, a robotic vehicle is used to drive two trajectories in the mapped area, of 4.6 and 12 km length, poles are extracted from the vehicle's sensor data as well, and the ability to derive the location correctly using pole matching is assessed.

2 ACQUISITION OF THE REFERENCE MAP

To obtain the reference map, a dense laser scan of a number of roads in Hannover, Germany, was acquired using the Steetmapper mobile mapping system (Kremer and Hunter, 2007) with a configuration of four scanners. Two scanners Riegl LMS-Q120 were pointing up and down at an angle of 20°, one was pointing to the right at an angle of 45°. Another Riegl LMS-Q140 was pointing to the left at an angle of 45°. The LMS-Q120 has a maximum range of 150 m and a ranging accuracy of 25 mm. All scanners were operated simultaneously at the maximum scanning angle

of 80° and scanning rate of about 10,000 points/s. Positioning was accomplished using IGI's TERRAcontrol GNSS/IMU system which consists of a NovAtel GNSS receiver, IGI's IMU-IId fiber optic gyro IMU operating at 256 Hz, an odometer, and a control computer. The scanned area contains streets in densely built-up regions as well as highway like roads. The total length of the scanning trajectory is 21.7 kilometers, 70.7 million points were captured. On average, each road meter is covered by more than 3,200 points.

From this point cloud, poles were extracted, such as traffic signs, traffic lights, and trees, to be used as landmarks. Poles have the property that they can be extracted quite reliably from the point cloud and are usually present in larger numbers in typical street corridors. The potential of poles for localization has been shown earlier by (Weiss et al., 2005) in the context of positioning along an intersection.

A relatively simple approach is used for finding and extracting poles, which is described in (Brenner, 2009a). The basic principle is to define poles as upright structures of a certain maximum diameter which do not contain other scanned points in their vicinity. This is controlled by two thresholds, the inner and outer (vicinity) radius. Due to attached structures such as signs or traffic lights, many actual poles would not qualify for this definition, as there would appear to be points in the vicinity. Therefore, analysis takes place in layers and a pole is accepted if the criterium is fulfilled for a certain number of layers. After being accepted, all points in the kernel are used for a least squares estimation of the pole center and the 2D coordinates of this center are inserted into the landmark map.

Using this algorithm, a total of 2,658 poles were found fully automatically, which is one pole every 8 meters on average. The results have been analyzed manually in (Hofmann and Brenner, 2009). Regarding the error rate, about 6% false positives were found on average, with higher rates of up to more than 30% in densely built-up areas. False positives are mainly due to occlusions, generating narrow bands of scan columns which have the same appearance as scanned poles.

3 EXTRACTING LANDMARKS USING A ROBOTIC VEHICLE

The second campaign was undertaken using 'Hanna', a robotic vehicle developed by the Real Time Systems Group (RTS) at the Institute for Systems Engineering at the Leibniz Universität Hannover, Germany. It was acquired along the trajectory of the first scan, however excluding highway parts because of the limited speed of the vehicle. Data were obtained in two runs, the first one with a length of 4.6 km, the second one with a length of 12 km.

'Hanna' uses four laser scanners SICK LMS 291-S05, which are mounted in pairs of two on rotary units so that full 3D scans are obtained. The maximum range is 30 m at a ranging accuracy of 60 mm. The vertical field of view is 90°, at 1° spacing. The horizontal field of view for the left scanner is from -210° to $+50^\circ$, and symmetric for the right scanner (taking 0° as the driving direction). It takes 2.4 s for a full rotation of the rotary unit. Using two scanners per unit, the time for a full scan is reduced to 1.2 s. However, for the experiments, only one scanner of one (the left) rotary unit was used. Clearly, this sensor setup is much less expensive and closer to a possible mass product, while being inferior to the Streetmapper scanner setup in terms of range, accuracy, and density.

For positioning, 'Hanna' uses a Trimble AgGPS 114 without differential corrections. A speed sensor at the gear and an angular encoder at the steering provide the odometer data. Both data sources were used in the experiments. Using the GPS data, 'standard' vehicle positioning was obtained. This is compared to a positioning based on odometer and steering angle data in conjunction with landmark matching (i.e., not using GPS at all). All data, including GPS positions and headings, odometer data and scanned point clouds, were time stamped and recorded for later processing.

Scan data is stored in terms of scan frames, where one frame is a horizontal sweep of any of the four scanners. One exterior orientation is given for each single frame. As the vehicle moves during the scan, scanpoints are motion compensated in such a way that the single, given orientation holds for all points of the frame.

While for Streetmapper data, pole extraction was performed in object space, this is much harder in case of the 'Hanna' datasets. The reason for this is that the vertical scan columns are relatively sparse. Therefore, in object space, it is almost impossible to tell if a column of stacked points is due to an actual pole or just a consequence of this low density. At the same time, combining several scans along the trajectory to obtain a higher density is also not trivial due to the relatively large positioning errors of the vehicle.

Therefore, the poles were extracted from single scans, using the raw data rather than the 3D point cloud. Since the points were recorded in succession, column by column, a neighborhood can be defined on the row and column indices. That is, analysis is performed on the depth image, indexed by the horizontal and vertical scan angles. One has to keep in mind, though, that this image is not an exact polar representation due to the forward movement of the vehicle during the scan. Poles are extracted by searching for vertical columns which have height jumps to both their left and right neighboring columns. Alternatively, successive pairs of columns are allowed with similar properties to account for poles in the foreground. The resulting point subsets are checked for a minimal height. A line is fitted to each point set and accepted based on the point residuals and the line orientation (which must be upright). Figure 1 shows the depth images and typical detection results.

In total, 2,250 poles were detected in 1,384 scans of the first run and 3,598 poles in 3,237 scans for the second run. In 35% (first run) to 40% (second run) of the scans, no poles were found at all. Figure 2 shows the distribution of the number of scans in which n ($0 \leq n \leq 9$) poles were found. As can be seen, the scans in which no poles were found is the largest group, and the percentage decreases quickly with increasing n. However, one has to keep in mind that still, (by accumulation) one or more poles were found in 60%-65% of the scans.

4 LANDMARK ASSOCIATION

Given the reference database of pole locations and the poles extracted from the 'Hanna' scans, the task is to associate and match poles in order to find the absolute position of the vehicle for every scan location. Of those two problems, association (of landmarks) is known to be the more critical step. Two subproblems may be differentiated, namely, finding the initial location and tracking the location (given the location of the previous scan). The initial location can be obtained using a match in the global pattern of poles, which turns out to be a standard 2D point pattern matching problem (Brenner, 2009b). This, however, requires to find a

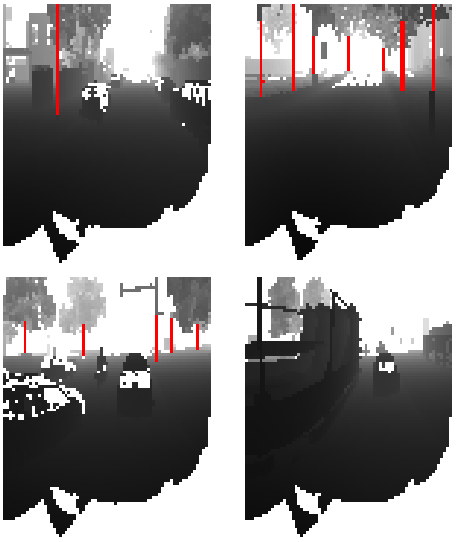

Figure 1: Typical pole detection results for 'Hanna' scans. Black is near, white is far (or error), and detected poles are marked in red. From left to right and top to bottom: typical inner city scene with buildings and trees to the left and right, alley scene, complex intersection along a road with several lanes, scene where possible poles are occluded by a large van. In all scenes (except the second), other cars can be seen in the foreground.

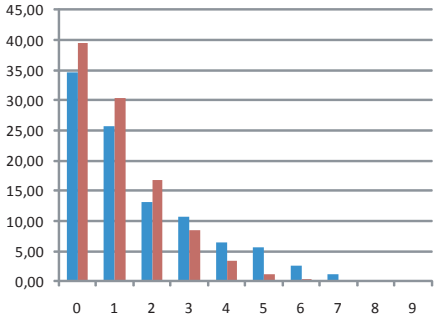

Figure 2: Histogram on the percentage of 'Hanna' scans (y axis) in which $0 \leq n \leq 9$ (x axis) poles were found. Left (blue) colums are for the first run, right (red) columns for the second run.

large enough subset of poles with sufficient accuracy, which is not easy to obtain, since it requires a good local positioning accuracy of the vehicle (remember that approximately 40% of the scans have no detected poles at all). Alternatively, GPS data can be used to find an approximate initial solution. Then, the second problem consists of tracking the location. A number of experiments were conducted regarding automatic tracking using poles (Brenner and Hofmann, 2010) which worked well in some cases but failed in others.

As it turned out, analysis of the errors is quite hard, since the matching of such long trajectories involves a huge number of automatically found associations, which cannot be judged easily from text protocols of the matcher. Therefore, it was decided to implement an interactive editor which allows to set, delete and visually explore the associations. This decouples the two tasks of association and matching, making the association a manual process. The interactive editor allows us to immediately see the effect of setting or deleting an association on the recovered trajectory. It allows to load different trajectories, such as a trajectory based on (absolute) GPS measurements and another, based on (relative) dead reckoning using odometry data only, and is able to switch between these trajectories instantly. It also can switch between different matching modules instantly, so that the effects of different matching strategies can be judged easily. Figure 3 shows two example views. Typically, a user will switch back and forth between different trajectores and matching modules while setting and deleting assignments. There is a convenient 'fence' mode for pole selection so that multiple scanned poles can be assigned to one reference pole at once.

Using this editor, 1,898 (84% of 2,250) poles of the first run and 2,223 (62% of 3,598) poles of the second run were associated. The manual assignment gave some insights into the situation. For example, it was expected that finding correct associations will be hard in an alley area, due to the high density of trees, leading to ambiguous situations. However, this was not the case. Since the editor allows to 'walk' along the trajectory, and making associations immediately has an effect on the trajectory, it was quite easy to incrementally find the associations along the alley. On the other hand, some seemingly 'easy' situations proved to be quite hard. There are a number of cases where trees, signposts, traffic- and streetlights stand very close, causing them to be missed in either the reference or the scan data. The editor also displays orthophotos with 0.4 m ground resolution, which help in some of the cases to disambiguate the situation. A key problem is that while dead reckoning gives a quite smooth trajectory, the heading drifts severly. This causes the position to drift quickly in situations where there is only a low density of poles. This large drift, combined with very close possible pole associations, causes a matching ambiguity. Even though being subjective in a sense, it is assumed that the manual assignments are correct in the vast majority of cases.

5 LANDMARK MATCHING

The matching algorithm has the task to track the current position, given previous positions and associated landmarks. It was not the intention to compute the best trajectory, using all available information (i.e., also future information), but rather, providing a (possibly real-time) solution using only current and history information. Normally, one would use an (extended) Kalman Filter or particle filter to obtain the estimate, using all available information from the vehicle sensors and the pole matches (Simon, 2006). However, in order to gain insight into the problem and the measurement and process noise, a least squares matching with a fixed 'history length' was used.

The matching filter works as follows. Given trajectories are processed, one exterior orientation (scan) after the other, in chronological order. If poles were found in a scan, they are added to the current history. The history is maintained up to a given maximum length only, i.e., after a new scan is added, it is checked if the maximum length is exceeded and poles from 'old' scans are removed, if necessary. Whenever poles have been added or removed, a 2D rigid transformation (3 parameters) is recomputed, using all pole associations in the current history and a (closed form) least squares estimation. This transformation yields the current position of the vehicle in the global coordinate system and also the residuals of the matched points.

Figure 4 shows the results for a small part of the trajectory of the second run (starting at the parking lot in the upper right corner of the image). In Fig. 4 (a), the GPS trajectory is shown, as delivered by 'Hanna' (without any pole matching involved). As can be seen, while the trajectory looks good in general, it hits the sidewalk on the parking lot and also after turning right into the street.

Figure 3: Example snapshots of the interactive association editor. Blue points are the current trajectory, yellow points are poles scanned by 'Hanna' (placed using the blue trajectory), green points are the reference poles as extracted from the Streetmapper data, and white lines mark associations between scanned and reference poles. Left: (raw) GPS trajectory with large residuals between associated poles. Right: trajectory based on dead reckoning and pole matching, using the same associations. The residuals were removed largely, leading to a much better trajectory.

Moving downwards from the upper left corner, it also leaves the road, first to the right (as seen from the vehicle's driving direction), later (in the lower half of the image) to the left. The deviation can also be seen from the difference vectors between the green reference poles and the yellow scan poles.

When applying the described filter, using a history length of 100 m, Fig. 4 (b) is obtained. Only the dead reckoning trajectory (and no GPS data) is used by the filter. Note that the trajectory (see the parking lot) does not start at scan number 0, but at scan 58, which is the first time the 2D rigid transformation can be computed, fixing the (relative) dead reckoning trajectory. While the trajectory looks better on the parking lot, it does not improve very much on the street going upwards. There, a certain 'sawtooth' pattern can be discerned. The reason for this behavior can be found in Fig. 4 (c), which shows the original dead reckoning trajectory without any matching applied. As can be seen, after leaving the parking lot, the vehicle shows a strong right drift, whereas later, after turning left into the second road, it shows a left drift.

To compensate for this behavior, the matching filter was redesigned to incorporate drift estimation. Instead of including drift to the parameters to be estimated (which would render the problem nonlinear), a 'multiple hypothesis' filter was used, which assumes a discrete set of possible heading drift values. For the experiments, 13 different drift assumptions were used, ranging from -0.34°/m to $+0.34^\circ$/m in equal steps. This amounts to maintaining 13 history buffers, which is, however, not computationally expensive. For each of the buffers, matching proceeds as described above. Then, the result with the smallest RMS error is taken. Figure 4 (d) shows the result of applying this filter to the dead reckoning trajectory, which outperforms both the result from the GPS as well as the result of the simple history filter. It can be seen that the filter is mostly able to compensate the drift, which leads to smaller RMS errors, being below 1 m except in a few cases. The drift, as determined by the filter, is shown in Fig. 5. The right and left turns in Fig. 4 (c) are properly reflected by the blue and orange / yellow parts in Fig. 5.

Figure 6 (left) shows the RMS error for the entire trajectory of run 2. It can be seen that there are several parts in the trajectory where the RMS error is above 1 m. For comparison, the number of assigned poles in the history is shown in Fig. 6 (right). It is especially low in the residential area in the upper part of the figure.

Figure 5: Drift correction applied by the multiple hypothesis filter for a part of run 2. Temperature scale, ranging from -0.34°/m (red, correction of left drift) to 0.34°/m (blue, correction of right drift).

As can be noticed, a large RMS error often correlates with low (assigned) point densities.

6 EVALUATION OF POSITIONING ACCURACY

The multiple history filter was applied to both runs. Note that, even with the multiple history modification, the filter is quite fast and processes each trajectory in approximately two seconds on a standard desktop PC. Unfortunately, no ground truth was available to assess the accuracy. Therefore, ortho images (0.4 m ground pixel resolution) were used to label manually all positions as being good or erroneous. For each single trajectory point, it was decided if it is located on a drivable area of the road, and, if it was a multiple lane road, if it is located on the correct lane. This method is of course more subjective than a rigorous comparison against a reference trajectory, and it also does not yield any assessment of the longitudinal error. Nevertheless, it is suitable to compare the matching results to the original GPS trajectory.

Figures 7 and 8 show the results of the manual assessment. It can be seen immediately that the number of erroneous positions is much larger in case of the GPS trajectory. Figure 8 shows some details in a densely built up area. At (A), GPS positioning fails badly, while matching is able to recover the position on

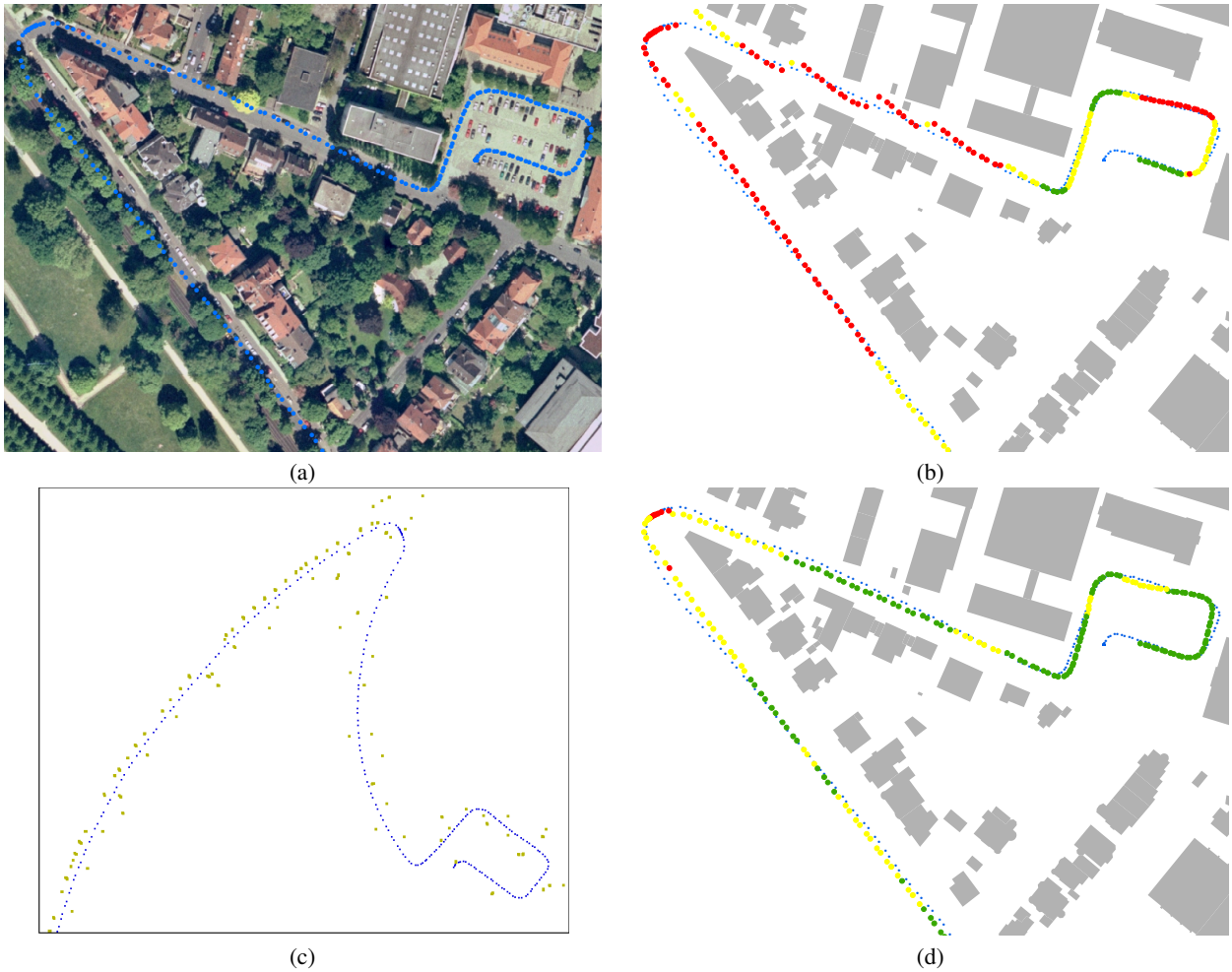

Figure 4: Example matching results for run 2. (a) Trajectory as obtained from the vehicle's GPS. (b) Result of applying the history based matching to the dead reckoning trajectory. Green marks RMS errors below 0.5 m, yellow between 0.5 m and 1 m, red above 1 m. The GPS trajectory is shown in blue. (c) Dead reckoning trajectory without any matching applied. (d) Result of applying the multiple history matching filter to the dead reckoning trajectory. Color coding identical to (b).

the road. (B) and (C) also show GPS problems, however, due to the low density of poles, the matching solution also drifts off the road. Finally, at (D), the 'Hanna' team missed the correct turn and entered a road which was not mapped by the Streetmapper van. With no matches at all, the trajectory follows the dead reckoning path (bottom image). However, also the GPS trajectory is mostly wrong in area (D).

In terms of quantitative results, 51.7% of the GPS trajectory points in run 2 were flagged as being erroneous, while this holds for only 13.0% of the matched trajectory (mostly due to the upper left part shown in Fig. 8). For run 1, the results are even better, where 39.6% errors were flagged in the GPS trajectory, but only 1.9% in the matched trajectory. This shows that the matching is able to largely improve localization whenever a sufficient number of landmark correspondences is available.

7 CONCLUSIONS AND OUTLOOK

In this paper, it has been shown that a landmark based localization is able to substantially outperform a GPS positioning solution in an extensive outdoor environment. Association of landmarks was done using an interactive tool, which provided insight into the problem and matching process.

There is lots of room for improvement. First, as some situations proved to be ambiguous, integrating the ground view into the editor would be valuable. Second, since the heading drift is severe, the matching algorithm has no chance to succeed when the density of poles is too low. Also, as the drift often changes abruptly at intersections, it is hard to estimate it (as compared to estimating the temperature drift of a gyro, which develops smoothly). Therefore, other methods have to be incorporated, such as dead reckoning based on matching successive scans (as done in standard SLAM approaches), which, however increases the computational requirements. Alternatively or additionally, other landmark features may be incorporated, such as planar patches. Finally, the matching algorithm itself has to be investigated in more detail, especially the RMS variations, and weak point configurations should be detected automatically. This should also yield tighter error bounds which are required to design a reliable automatic association procedure.

ACKNOWLEDGEMENTS

We thank Matthias Hentschel from the Real Time Systems Group of the Institute for Systems Engineering of the Leibniz Universität Hannover for acquiring and providing the RTS 'Hanna' scans.

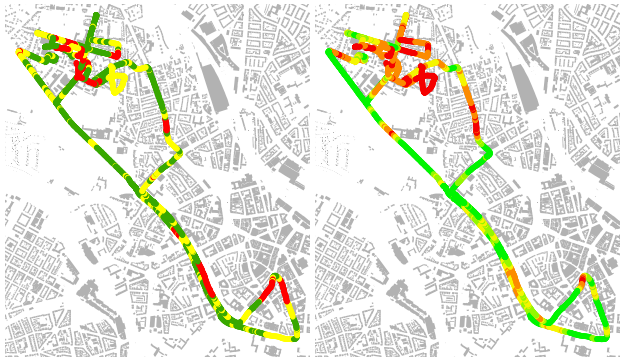

Figure 6: Full trajectory of run 2. Left: RMS error, color coding as in Fig. 4(b). Right: Number of assigned poles in the history buffer. Red: 0-5, orange: 6-10, yellow: 11-15, light green: 15-20, dark green: more than 20.

Figure 7: Results of the manual assesment for run 2 (12 km). Bold trajectory points mark erroneous positions. Left: GPS trajectory. Right: Result of matching.

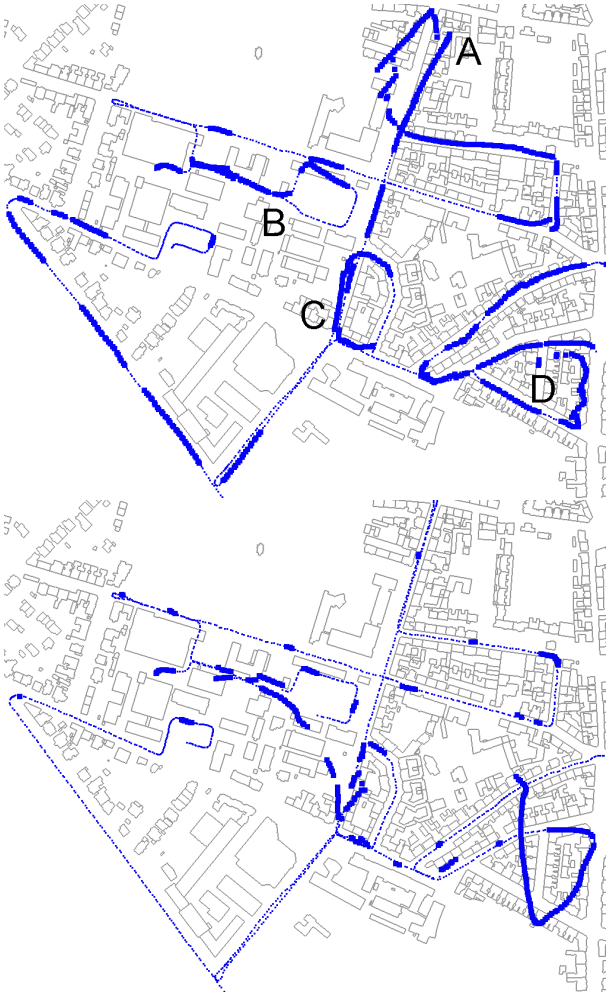

Figure 8: Detail view of upper left part of Fig. 7. Top: GPS trajectory. Bottom: Result of matching. See text for an explanation of A-D.

REFERENCES

Borrmann, D., Elseberg, J., Lingemann, K., Nüchter, A. and Hertzberg, J., 2008. Globally consistent 3D mapping with scan matching. Journal of Robotics and Autonomous Systems (JRAS) 56(2), pp. 130–142.

Brenner, C., 2009a. Extraction of features from mobile laser scanning data for future driver assistance systems. In: Advances in GIScience, Lecture Notes in Geoinformation and Cartography, Springer Berlin Heidelberg, pp. 25–42.

Brenner, C., 2009b. Global localization of vehicles using local pole patterns. In: Pattern Recognition, LNCS 5748, Lecture Notes in Computer Science, Springer, pp. 61–70.

Brenner, C. and Hofmann, S., 2010. Evaluation of automatically extracted landmarks for future driver assistance systems. In: Proc. 14th International Symposium on Spatial Data Handling, Hong Kong, May.

Burgard, W. and Hebert, M., 2008. Springer Handbook of Robotics. Springer, chapter World Modeling, pp. 853–869.

EDMap, 2004. Enhanced digital mapping project final report. Technical report, United States Department of Transportation, Federal Highway Administration and National Highway Traffic and Safety Administration.

Hofmann, S. and Brenner, C., 2009. Quality assessment of automatically generated feature maps for future driver assistance systems. In: Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS '09, ACM, New York, NY, USA, pp. 500–503.

Kremer, J. and Hunter, G., 2007. Performance of the streetmapper mobile lidar mapping system in real world projects. In: Photogrammetric Week 2007, Wichmann, pp. 215–225.

Kukko, A., Andrei, C.-O., Salminen, V.-M., Kaartinen, H., Chen, Y., Rönnholm, P., Hyyppä, H., Hyyppä, J., Chen, R., Haggrén, H., Kosonen, I. and Čapek, K., 2007. Road environment mapping system of the finnish geodetic institute – FGI roamer. In: IAPRS (ed.), Proc. Laser Scanning 2007 and SilviLaser 2007, Vol. 36 Part 3/W 52, pp. 241–247.

Pandazis, J.-C., 2002. NextMAP: Investigating the future of digital map databases. In: e-Safety Congress, Lyon.

Simon, D., 2006. Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches. John Wiley & Sons.

Thrun, S., Burgard, W. and Fox, D., 2005. Probabilistic Robotics. The MIT Press, Cambridge, Mass.

Weiss, T., Kaempchen, N. and Dietmayer, K., 2005. Precise ego localization in urban areas using laserscanner and high accuracy feature maps. In: Proc. 2005 IEEE Intelligent Vehicles Symposium, Las Vegas, USA, pp. 284–289.