# Localization Using Automotive Laser Scanners and Local Pattern Matching

Alexander Schlichting and Claus Brenner<sup>1</sup>

Abstract—Autonomous driving requires vehicle positioning with accuracies of a few decimeters. Typical low-cost GNSS sensors, as they are commonly used for navigation systems, are limited to an accuracy of several meters. Also, they are restricted in reliability because of outages and multipath effects. To improve accuracy and reliability, 3D features can be used, such as pole-like objects and planes, measured by a laser scanner. These features have to be matched to the reference data, given by a landmark map. If we use a nearest neighbor approach to match the data, we will likely get wrong matches, especially at positions with a low initial accuracy. To reduce the number of wrong matches, we use feature patterns. These patterns describe the spatial relationship of a specific number of features and are determined for every possible feature combination, separated in reference and online features. Given these patterns, the correspondences of the measured features can be determined by finding the corresponding patterns in the reference data.

We acquired reference data by a high precision Mobile Mapping System. In an area of  $2.8 \text{ km}^2$  we automatically extracted 1390 pole-like objects and 2006 building facades. A (second) vehicle equipped with an automotive laser scanner was used to generate features with lower accuracy and reliability. In every scan of the laser scanner we extracted landmarks (poles and planes) online. We then used our proposed feature matching to find correspondences. In this paper, we show the performance of the approach for different parameter settings and compare it to the nearest neighbor matching commonly used. Our experimental results show that, by using feature patterns, the rate of false matches can be reduced from about 80 % down to 20 %, compared to a nearest neighbor approach.

## I. INTRODUCTION

For autonomous driving as well as future driving assistance systems highly accurate localization is essential. Since a GNSS solution is limited in accuracy and reliability, additional measurements are required. Most self driving cars contain (multilayer) automotive laser scanners, as they are a conventional sensor for obstacle detection and collision avoidance [1]. In addition, these sensors can be used to detect landmarks, to improve the ego position given by a GNSS/INS system. In comparison to 3D laser scanners, the advantages of automotive systems are their relatively low price and compact size. They can be integrated in the front bumper as shown in [2].

An important part of localization using features is the digital map, which contains the reference data. In the future, this data should be available for every road. Thus, an automatic



Fig. 1: Measurement principle of the used automotive laser scanner.

feature segmentation method is required. For pole-like objects, we use the method presented in [3]. It automatically extracts poles from dense 3D laser point clouds. We also extract building facades by segmenting planar regions. The data is acquired using a Riegl VMX-250 LiDAR Mobile Mapping system. Another possibility to obtain reference data is to continuously use the measurements made by the user vehicles and to combine them in a Simultaneous Localization and Mapping (SLAM) approach. The online features (poles and planes) are segmented from a multilayer automotive laser scanner with an installation height of about 0.5 meters above ground (see figure 1). Particularly because of the narrow vertical beam width, we get a lot of false positive feature detections. Furthermore the low laser scanner installation height reduces the number of detected features. These characteristics have a negative impact on the matching of the detected features to their references. The matching becomes even more complex, if our initial position is inaccurate. Hence, we use local feature patterns to analyze the spatial relations of the detected poles and planes compared to the relations of the reference features. Based on successfully matched features, the transformation of our ego position can be calculated. We also analyze different fixed combinations of poles and planes for defining the patterns.

The paper is organized as follows. Section II gives a review of related work. In section III the acquisition of the reference features and the online detected features is described. The feature patterns approach is presented in section IV, with an experimental evaluation in section V. Finally, in section VI conclusions are drawn. The focus of our work is on the online feature extraction in combination with local feature patterns for feature association.

# II. RELATED WORK

Especially since the DARPA Grand Challenges in 2004 and 2005 and the DARPA urban challenge in 2007, autonomous driving has become an important research field. In most cases, the vehicle is localized by a combination of an INS/GNSS sensor and 2D features, measured by a laser scanner or a camera. One of the first autonomous driving

<sup>&</sup>lt;sup>1</sup>Authors are with the Institute of Cartography and Geoinformatics, University of Hannover, 30167 Hannover, Germany alexander.schlichting@ikg.uni-hannover.de, claus.brenner@ikg.uni-hannover.de

cars on highways was a vehicle developed by Pomerleau and Jochem [4]. They detected lane markers by video images to get the local vehicle position on the street and to estimate the current road curvature. Recently, Lategahn et al. [5] used mono cameras in combination with an inertial measurement unit (IMU) for global vehicle localization in urban areas. Landmarks were detected in the images and compared to a reference landmark map. Although they show, that the algorithm works in the morning and in the evening, this approach is restricted by illumination conditions and will probably not work at night.

The vehicle of the 'Stadtpilot' (city pilot) uses a LiDAR sensor to detect lane markers by their high reflectivity [6]. Then these features are matched to a digital map to correct the position given by an INS/GNSS combination improved by Real Time Kinematic (RTK) GPS. Levinson et al. [7] are using LiDAR sensors to extract a 2D surface image of the ground. They correlate the extracted images to a given map and localize the vehicle relative to this map using a particle filter. One disadvantage of using 2D features is the dependency on weather conditions. Leaves, snow or water on the ground can complicate the estimation of the ego position. 3D features measured by a laser sensor, as proposed in [8], are mostly independent on weather conditions and illumination. The measurement accuracy may decrease at snowfall or rain, but a localization is still possible.

The potential of a 3D landmark based localization approach was shown in [9], where 3D laser scanners in combination with a high accurate GNSS/INS system are used to extract poles. In [10] the localization accuracy was improved by poles extracted from four 3D laser scanners and matched to a reference data set. The matching uses an algorithm similar to random sampling consensus (RANSAC), which in this case works fine because of a high count of reliable features, guaranteed by laser scanners mounted on the top of the vehicle.

If we use an automotive laser scanner with only a few measurement layers, the data will contain overall a smaller number of measured features and a higher false positive rate. Matching the features, using a nearest neighbor approach or the algorithm presented in [10], won't work sufficiency. Alternative approaches are proposed in [11] and [12]. Both of them suggest local point patterns to estimate global vehicle localization without any prior information about its position and orientation. Our method is based on [12], whereas we do not restrict the feature to point shapes but include lines, given by extracted vertical planes.

# **III. DATA ACQUSITION**

## A. Reference Feature Extraction

The point cloud from which we extracted the reference data was measured by a Riegl VMX-250 LiDAR Mobile Mapping System. The system consists of two scanners, each with a scanning rate of 300,000 points/s [13]. The maximum range is 200 meters, the ranging accuracy is ten millimeters. For localization, we used an Applanix POS LV GNSS/IMU



Fig. 2: Distance measurements for one scan line of the laser scanner.

system. The GNSS measurements were improved by correction data of the Satellite Positioning Service SAPOS. Due to the fact that we do not need to know the position online, it is possible to generate a highly accurate trajectory by post processing. For pole segmentation we used the algorithm presented in [3]. We extracted 1390 pole-like objects in our test region. Because of the assumption that they are upright, we only saved the coordinates of the pole center point, which we determined by using a cylinder estimation method from the C++ Point Cloud Library [14].

In addition to cylindrical objects, we extracted planes, namely building facades. Therefore we used a region growing algorithm, as it is presented in [15]. As a criterion of homogeneity the Euclidean 3D distance and the local normal vector is used. We set the minimal number of points to 2000 and the minimal plane height to three meters, in order to prevent the detection of cars and hedges. We extracted 2006 building facades in the test area. For these planes we saved the plane parameters, given by the plane normal vector and the distance to the coordinate origin, and the coordinates of the peripheral points to our feature map.

# B. Online Feature Extraction

For online feature detection, we used the measurements of an automotive multilayer laser scanner, like the ibeo LUX [16]. This scanner measures four layers with an overall vertical field of view of 3.2 degrees, a horizontal resolution of 0.25 degrees, and a scanning frequency of 25 Hertz. The measurement range is 200 meters. The sensor is mounted at the front of the vehicle, about half a meter above the ground. The measurement setup is outlined in Figure 1.

Figure 2 shows the distance measurements for one scan line. Continuously varying measurements are a typical sign for planar regions in the scene. Using a distance jump criterion, we detected the borders of such segments. Since they represent only a small subset of the object, they cannot be assigned reliably. Furthermore, with one scan we only have 2D information of the shape of the scanned object. Thus, the scan lines have to be grouped into one large segment. For this purpose, we save the 3D points and the corresponding plane parameters every time we detect a planar segment. Since we know about the vertical plane direction, we set the z-direction of the normal vector to zero. In the next step, we match the plane to planes we have detected



Fig. 3: Detected pole in the 3D point cloud. The points from the Riegl VMX-250 measurement systems are colored green, the points from the automotive laser scanner are colored blue.

before, by comparing the peripheral points and the normal vectors of the planes. The peripheral points are analyzed by their Euclidean horizontal distance, to check whether the planes overlap each other. If the planes overlap and the angle of the planes is below a specific threshold, which is set to 20 degrees, the planes are assigned to each other and the plane parameters are updated by the new values. Otherwise, it is saved as a new plane. The goal of segmenting features like planes is to correct the current ego position. Because the measured plane parameters depend directly on the uncorrected ego position, we do not average the position of the measured planes but rather use the parameters given by the last measured scan. In addition, we implemented a score to see how often a plane is detected and matched. If we have detected a plane for a specific number of times (here, 15 times), we use it in the matching algorithm.

Similar to the plane extraction and matching algorithm, we detected poles. A pole is a freestanding object. For that reason, the measured distances of the laser beams show two depth jumps along the scan line, at the left and right edge of the pole. Another criterion is that because of the small diameter of poles, the number of successive measurements without distance jumps must be small. Using these features, we detect poles in each scan line. In a second step, we calculate the global coordinates of the pole. If there already is a pole in a specific search radius, we use the position provided by the current measurement. The score is increased by  $\frac{1}{d^2}$ , where *d* is the Euclidean 2D distance of the measured pole to the given pole. The initial pole score is set to 1.

To reduce the number of false positives, we try to avoid features on the street. This condition is detected by the following criteria:

• The distance of at least one vehicle position to the pole must be below a certain threshold. In our experiments the distance is set to ten meters. In addition for this position the angle between the feature (pole or plane center point) and the ego position must be in a specific range (see figure 4). Here, the pole angle range expediently is from -135 to -45 degrees and from 45 to 135 degrees.



Fig. 4: Reducing features by an angle range of 45 degrees and a maximum vehicle distance of ten meters (blue) as well as a minimum distance of two meters (red). Valid features are colored green, invalid features red. The trajectory is colored blue.

• Features whose distance to any vehicle position is lower than two meters are not considered.

In the future this validation can be done by using further information, for example by segmented lane markers to classify the street.

Figure 3 shows a detected pole-like object in the point cloud. The green colored points, measured by the Riegl VMX-250 Mobile Mapping system, are well distributed over the whole object. In contrast, the points measured by the automotive laser scanner (blue) only occur in a small region at the lower end of the pole, due to the small vertical field of view of 3.2 degrees.

# **IV. FEATURE PATTERNS**

Given a set of online detected features and a landmark map, the poles and planes have to be matched to their references to improve the current ego position. The matching uses the local pattern formed by single features. For a combination of detected features, a descriptor is computed and compared to reference descriptors. If a similar descriptor is found, the measured features are matched to the associated reference landmarks. The transformation then is defined by the coordinate differences of the detected poles to the reference poles. Basically we use the method presented in [12], but added planes to the matching algorithm and reduced the number of analysed patterns, which we determine online.

### A. Feature Patterns Descriptor

The original descriptor presented in [12] is designed as follows. For a set of  $k_{pole} \geq 2$  poles the first value of the descriptor D is determined by the largest distance between any two poles in the current pole set. This diameter also defines the x-axis of the local pattern coordinate system, with the y-axis perpendicular to it. The system orientation is fixed by specifying that the extension of the remaining  $k_{pole} - 2$  poles in +y is larger than in -y. The remaining poles are sorted lexicographically by their  $(x_i, y_i)$  values. These values are added to the descriptor, reduced to the origin of the local coordinate system. This leads to a dimension  $dim = 2 \cdot k_{pole} - 3$  for each descriptor.

In contrast to [12] we also handle planes to improve the



Fig. 5: Feature patterns principle for  $k_{pole} = 3$  and  $k_{plane} = 1$ . The reference planes and poles are colored blue, the measured features green. The links between the features represent the descriptor values, the resultant reference descriptor (blue) is  $D\{d, x_1, y_1, x_2, y_2, \alpha\}$ .



Fig. 6: 3D plot of all local reference descriptors in the testing area for  $k_{pole} = 3$  and  $k_{plane} = 0$ .

feature matching. We assume the planes to stand vertical on the ground. First the  $k_{plane}$  planes are sorted in lexicographic order by their center point, which we define by the average coordinates of the two peripheral plane points. In the next step the planes are added to the descriptor one after another in the following way. First we add the center points the same way, as we add poles to the descriptor. To represent the plane orientation, we further add the angle between the xaxis and the plane to the descriptor. The pattern generation is outlined in figure 5. By adding planes to the descriptor in this way, the dimension is now  $dim = 2 \cdot k_{pole} + 3 \cdot k_{plane} - 3$ .

Alternatively to the plane center point coordinates one could use the minimal distance of the plane to the origin of the coordinate system, which is calculated using the plane parameters. This would reduce the descriptor dimension to  $2 \cdot k_{pole} + 2 \cdot k_{plane} - 3$  but also cause several problems. One problem is that the uniqueness of the descriptors would decrease. If some separated building facades are arranged in a row with a similar orientation, all of these planes would have a similar distance to the origin. Furthermore the accuracy of this value would be influenced by the plane orientation accuracy.

# B. Matching Process

In [12] for every reference pole  $p_i$  all possible combinations of k-1 poles from the neighborhood  $P_i$  of  $p_i$  are selected. For these poles a Descriptor D is computed and stored in a database with the key D and the value *i*. The database can be used to match the features to their references as follows. Figure 6 shows all local reference descriptors plotted in 3D space. If for a given scene at least k poles  $p_i$  have been detected, the Descriptor  $D_i$  of k randomly selected poles from this scene can be computed. A set of possible solutions of the correct pole correspondences is provided by the database. This step can be repeated until there is only one solution remaining. A reference descriptor is marked as possible solution, if their values are nearly identical. This means, that all absolute value differences must be smaller than the respective feature measurement accuracy factors  $e_{pole}$ ,  $e_{x/y,plane}$  and  $e_{\alpha,plane}$ . The pole measurement error  $e_{pole}$  can be set to some decimeters as proposed in [12]. Unlike the pole center point, the plane center point is not only influenced by the measurement accuracy. It is not guaranteed, that the laser scanner measures the full plane extension. Occlusions may lead to wrong peripheral points and as a result to a shifted plane center point (see figure 5). Therefore we set  $e_{plane}$  to a much higher value of ten meters. We varied the pole measurement accuracy  $e_{pole}$  and the plane angle measurement accuracy  $e_{\alpha, plane}$  and compared the results as shown in section V.

In this work for every possible solution a translation  $T_k$  is determined by the 2D coordinate differences of the current pole set to the respective reference pole coordinates and saved in a list. This step is repeated iteratively. The translations contain a score. If the list already contains a similar translation, the values are averaged, weighted by the score and the score gets increased. Similarity is identified by specifying a threshold for the translation differences. Here the threshold was set to 0.5 meters. The matching step is done, if the number of iterations exceeds a certain limit or if the translation with the highest count is unique. For this work we defined a translation is higher than five. Note that this fixed threshold is only for experimental purpose and can be set dynamically in further works.

The time intensive step of creating the reference patterns database can be done offline. As a disadvantage, the data may become very large. In the case of an autonomously driving car, the database has to be stored on the vehicle's processing unit with a limited memory size or on a server, with a guaranteed permanent and fast user internet access. To avoid this problem, we compute the descriptors online. To reduce data and as a consequence to speed up computation, we only take landmarks in the neighborhood of the detected features.

As described in section III-B, the feature positions are directly influenced by the current errors in localization. These errors are not constant over time. Hence, we have to restrict the features of a descriptor to a small extend, which we set to 50 meters around our current ego position.

## V. EXPERIMENTAL RESULTS

In this section we will evaluate experimental matching results of a test region in Hannover, Germany. Along a trajectory of about 5.5 km length we applied the feature pattern algorithm 935 times at a frequency of one determination per second. Our measurement vehicle was equipped with an OXTS RT3000 GNSS/INS module with differential GNSS and a localization accuracy of a few decimeters [17]. The laser scanner is similar to a ibeo LUX [16] with four scanning layers and a vertical field of view of 3.2 degrees (see figure 1). The scanning frequency was set to 25 Hertz yielding to an angular resolution of 0.25 degrees. The specified distance resolution is four centimeters.

We use two indicators to evaluate our feature matching method. First we analyze the matching success rate, which is given by the number of successfully matched patterns which lead to the correct position, relative to the total number of positions (completeness). Secondly, the percentage of correct matches in comparison to matches with wrong transformation results is an indicator of the matching correctness. The best result would be a large number of correct matches and a low number of false positives, i.e. a large completeness and correctness. Otherwise, wrong matches would lead to errors in a subsequent filter correction step. These results are compared to a simple nearest neighbor approach, where the measured features are matched to the reference features with the lowest Euclidean 2D distance.

Because we are using a highly accurate positioning system to evaluate the matching method, we expect the length  $\sqrt{\Delta x^2 + \Delta y^2}$  of correct translation vectors to be nearly as small as the localization accuracy. Nevertheless we could set the threshold above which a translation is marked as wrong to one meter, as we expect poles to have no neighbor poles within this radius.

Exemplary maps of the trajectory, with successfully matched features in green and false matches in red, are shown in figure 7. Tables I-III show the matching results for varied strict matching conditions, separated for the different number of involved features. One can see that the number of successfully matched features increases minimally with a higher pattern parameter tolerance, which means that high tolerances yield to more trajectory points with a correct ego position. In contrast, the percentage number of correct matches decreases with a higher tolerance, from which follows that low tolerance values have a positive effect on localization reliability. Table I shows the results for the case, when the descriptor uses three poles and no plane. Replacing a pole feature by a plane feature  $(k_{pole} = 2 \text{ and } k_{plane} = 1)$ effects a less correct matching and a comparable matching success rate. The best results for the number of correct matches are at  $e_{pole} = 0.30 \ m$  and  $e_{\alpha, plane} = 15 \ deg$  with  $k_{pole} = 3$  and  $k_{plane} = 1$ . In contrast, the matching results of a nearest neighbor approach are 10 % for the matching rate and 18 % for the number of correct matches. Compared to this, the percentage of correct matches in every case is strongly increased. On the other hand, the completeness is decreased. Assuming a correct matching and highly accurate reference data, previous experiments have shown that a localization accuracy of six centimeters is possible, using poles and planes measured by an automotive laser scanner [18].

The average segmentation running time is 28 ms (poles) and 161 ms (planes) for every trajectory point in a one second interval, which corresponds to 25 scanner measurements. The

TABLE I: Matching results for  $k_{pole} = 3$  and  $k_{plane} = 0$ .

Parameters		Matching rate	Correct matches
$e_{pole}$ [m]	$e_{\alpha,plane}$ [deg]	(completeness)	(correctness)
0.20	10	4 %	73 %
0.30	15	6 %	83 %
0.50	20	7 %	73 %
1.00	45	7 %	56 %

TABLE II: Matching results for  $k_{pole} = 3$  and  $k_{plane} = 1$ .

Parameters		Matching rate	Correct matches
$e_{pole}$ [m]	$e_{\alpha,plane}$ [deg]	(completeness)	(correctness)
0.20	10	2 %	70 %
0.30	15	3 %	89 %
0.50	20	3 %	64 %
1.00	45	3 %	55 %

TABLE III: Matching results for  $k_{pole} = 2$  and  $k_{plane} = 1$ .

Parameters		Matching rate	Correct matches
$e_{pole}$ [m]	$e_{\alpha,plane}$ [deg]	(completeness)	(correctness)
0.20	10	6 %	52 %
0.30	15	6 %	44 %
0.50	20	6 %	38 %
1.00	45	6 %	32 %

average running time of the pattern analysis (including the reference patterns generation) is 7 ms for  $k_{pole} = 3$  and  $k_{plane} = 0, 23$  ms for  $k_{pole} = 3$  and  $k_{plane} = 1$  and 11 ms for  $k_{pole} = 2$  and  $k_{plane} = 1$ . The feature filtering step, to prevent false positives, requires 70 ms per calculation on average. The evaluation was performed on a Windows 7 64 bit system with a 3.70 GHz CPU and 8 GB main memory.

# VI. CONCLUSION

In this paper, we have presented and evaluated an approach to match features measured by a low-cost sensor to high quality reference data by local feature patterns. Using the proposed matching algorithm, the number of erroneously matched features can be decreased from 82 % to 11 % in comparison to a nearest neighbor approach. For future works, we suggest to combine different descriptors with a different number of features to stabilize the feature matching. Because of the overall low success rate, landmarks extracted from automotive laser scanners are not capable as a standalone solution for ego localization but recommendable to improve any given ego localization. The running time evaluation shows, that the presented approach is real-time capable.

An important factor for a reliable feature matching is the number of outliers. Herein outliers are reduced by several constant thresholds, regarding to the local position of the features in the scene. In future works, these thresholds can be replaced by a scene classification. For example additional sensors like cameras or the intensity values of a laser scanner can be used to detect lane markers and in this context to classify the road surface. Assuming that there are no features on the road, we can reduce the number of outliers. The



(d) Nearest neighbor.

Fig. 7: Trajectory with correct matches in green and wrong matches in red for  $e_{pole} = 0.30 \ m$  and  $e_{\alpha, plane} = 15 \ deg$ compared to a nearest neighbor approach. The blue symbols represent trajectory points where no matching did succeed [19].

detected lane markers or any other features also can be used to extend the local pattern matching algorithm.

#### REFERENCES

- [1] K. C. Fuerstenberg, K. C. Dietmayer, and V. Willhoeft, "Pedestrian recognition in urban traffic using a vehicle based multilayer laserscanner," in Intelligent Vehicle Symposium, 2002. IEEE, vol. 1. IEEE, 2002, pp. 31-35.
- [2] G. Stanek, D. Langer, B. Muller-Bessler, and B. Huhnke, "Junior 3: A test platform for advanced driver assistance systems" in *Intelligent* Vehicles Symposium (IV), 2010 IEEE. IEEE, 2010, pp. 143-149.
- [3] C. Brenner, "Extraction of features from mobile laser scanning data for future driver assistance systems," in Advances in GIScience. Springer, 2009, pp. 25-42.
- [4] D. Pomerleau and T. Jochem, "Rapidly adapting machine vision for automated vehicle steering," IEEE expert, vol. 11, no. 2, pp. 19-27, 1996.
- [5] H. Lategahn, M. Schreiber, J. Ziegler, and C. Stiller, "Urban localization with camera and inertial measurement unit," in Intelligent Vehicles Symposium (IV), 2013 IEEE. IEEE, 2013, pp. 719-724.
- [6] T. Nothdurft, P. Hecker, S. Ohl, F. Saust, M. Maurer, A. Reschka, and J. Bohmer, "Stadtpilot: First fully autonomous test drives in urban traffic," in Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on. IEEE, 2011, pp. 919-924.
- [7] J. Levinson, M. Montemerlo, and S. Thrun, "Map-based precision vehicle localization in urban environments." in Robotics: Science and Systems, 2007.
- [8] T. Weiss, N. Kaempchen, and K. Dietmayer, "Precise ego-localization in urban areas using laserscanner and high accuracy feature maps," in Intelligent Vehicles Symposium, 2005. Proceedings. IEEE, IEEE, 2005, pp. 284-289.

- [9] S. Hofmann and C. Brenner, "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. ACM, 2009, pp. 500-
- [10] C. Brenner and S. Hofmann, "Evaluation of automatically extracted landmarks for future driver assistance systems," in Advances in Spatial Data Handling and GIS. Springer, 2012, pp. 169–181. [11] D. Ronzoni, R. Olmi, C. Secchi, and C. Fantuzzi, "AGV global
- localization using indistinguishable artificial landmarks," in Robotics and Automation (ICRA), 2011 IEEE International Conference on. IEEE, 2011, pp. 287-292.
- [12] C. Brenner, "Global localization of vehicles using local pole patterns," in Pattern Recognition. Springer, 2009, pp. 61-70.
- [13] "RIEGL VMX-250," RIEGL Laser Measurement Systems GmbH, Tech. Rep., 2012. [Online]. Available: http://www.riegl.com
- Library. [14] Point Cloud Cylinder model segmentation. [Online]. Available: http://pointclouds.org/documentation/tutorials/ cylinder\_segmentation.php
- [15] R. Adams and L. Bischof, "Seeded region growing," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 16, no. 6, pp. 641-647, 1994.
- [16] "ibeo Lux," Ibeo Automotive Systems GmbH, Tech. Rep., 2013. [Online]. Available: http://www.ibeo-as.com
- [17] "RT3000 Inertial and GPS Navigation System - Datasheet," Oxford Technical Solutions Ltd, Tech. Rep., 2013. [Online]. Available: http://www.oxts.com
- [18] A. Schlichting and C. Brenner, "Genauigkeitsuntersuchung zur Lokalisierung von Fahrzeugen mittels Automotive-Laserscannern," in Publikation der Deutschen Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation e.V. - Vorträge der 34. Wissenschaftlich-Technischen Jahrestagung der DGPF. DGPF, 2014.
- OpenStreetMap. Map of Hannover-Badenstedt, Germany. [Online]. [19] Available: http://www.openstreetmap.org/node/255606172