2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

Global Vehicle Localization by Sequence Analysis Using LiDAR Features Derived by an Autoencoder

Alexander Schlichting and Udo Feuerhake

Abstract— **Global vehicle localization is normally done by GNSS sensors. In case of GNSS outages, such as in urban canyons or tunnels, highly automated cars cannot localize without an initial known position. In this paper we propose a method for absolute localization in a city based on only one 2D laser scanner.**

In out method, localization is done by matching vertical scan lines captured by a 2D laser scanner mounted on the vehicle with scan lines derived from a reference point cloud of the environment. We use a neural network to derive significant features describing the shape of the scan lines. Every scan line of a reference data set is labeled with a specific cluster-id using a k-means algorithm and stored in a reference graph. The same k-means algorithm is used to label the single scan lines of a test drive. The localization is done via a sequence mining approach, where a sequence with a specific length is matched to the position with the highest correlation in the reference sequence.

In our experiments we analyze the effect of several parameters, including the number of features and sequence length. The results show that the algorithm performs with an accuracy of about 1.4 m and a completeness of up to 99%. Even if the input scan is represented by only ten features, the results are betten than those obtained by using the whole range scan in the localization step.

I. INTRODUCTION

Vehicle localization is an important issue for driver assistance systems and especially fully automated vehicles. To guarantee a highly accurate and reliable localization most approaches use several sensors in one system. A filter approach is used to combine the measured data of sensors, like inertial systems, GNSS sensors, cameras, radar and laser scanners. Normally the absolute position of the system is given by the GNSS sensor.

Due to the fact that vehicle manufacturers expedite the automation of vehicles step by step, localization is a hot research topic with many publications and promising results in the last few years. Trigger events for this trend have been the DARPA Grand Challenge in 2004 and the DARPA urban challenge 2007. Since GNSS signals are not always available, e.g. in urban canyons and tunnels, localization techniques that can work independent of GNSS have received a great deal of attention.

Levinson and Thrun [1] have used a 3D laser scanner mounted on the roof of the car to record an intensity image and compare it to a digital map by image correlation. The vehicle of the 'Stadtpilot' (city pilot) project uses a similar approach [2]. Ziegler et al. [3] combine a lane marker detection approach with a 2D point feature detection. In [4] and [5] 3D landmarks, namely pole-like objects, are used for localization. All the listed approaches have in common, that an initial absolute position has to be known as a prior, normally given by a GNSS sensor.

Khoshelham et al. [6] [7] are combining visual odometry and inertial data to bridge gaps in the GNSS data. Their method is suitable for short periods of time. However due to drift effects this method is not long-time stable.

The Dutch navigation and mapping company TomTom introduced a localization system called RoadDNA, that provides a reliable and accurate localization based on LiDAR (Light detection and ranging) data [8]. The approach is described in Li et al. (2016) [9]. A reference map is generated by projecting a 3D point cloud onto planes that are perpindicular to the road surface. In the online step they match a map piece of in this case 50 m length to the reference map. By adding low-cost GPS and IMU measurements like described in [10] they achieve a localization accuracy of 0.35 m in lateral and 0.40 m in longitudinal direction.

However, in case of GNSS outages and without any prior information, the above methods fail. GNSS outages may appear because of jamming [11] [12] or spoofing [13] signals or even because the system operator deactivates the system. Due to effects like multi-path and poor satellite visibility, especially in urban areas with high buildings, using only GNSS measurements can lead to large errors and gaps in the data. A complementary system is needed to guarantee a reliable absolute localization, even in case of GNSS outages. The determination of absolute vehicle position can be interpreted as a place recognition problem or a lost robot problem in robotics. Bosse and Roberts [14] correlate feature histograms to match laser scans to a reference in unstructured outdoor environments. Tipaldi et al. [15] use so called geometrical FLIRT phrases to perform a 2D range data matching. The matching is done by a bag-of-words approach. Himstedt et al. [16] use Geometrical Landmark Relations (GLARE). GLARE transforms 2D laser scans into pose invariant histogram representations and performs an approximate nearest neighbor search to match the data to a reference. They could achieve a recognition rate of 93% along a 6.5 km outdoor trajectory.

In this paper we introduce a method that achieves a recognition rate of up to 99% along an even longer trajectory in an urban area without using GNSS measurements. The absolute position is determined by matching scan lines captured by

A. Schlichting and U. Feuerhake are with the Institute of Cartography and Geoinformatics, Leibniz Universität Hannover, 30167 Hanover, Germany (e-mail: alexander.schlichting@ikg.uni-hannover.de; udo.feuerhake@ikg.uni-hannover.de).

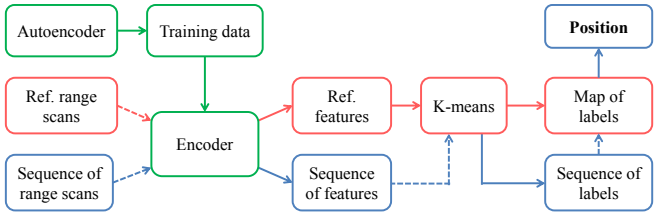


Fig. 1: Flowchart of our approach: first an autoencoder is trained using range scans of our reference data as training data. After the training step, the encoder is separated from the autoencoder and used to generate features from the laser scans. A reference graph of clustered labels is generated. Finally, the labeled sequence of range scans is matched to the graph to localize the vehicle.

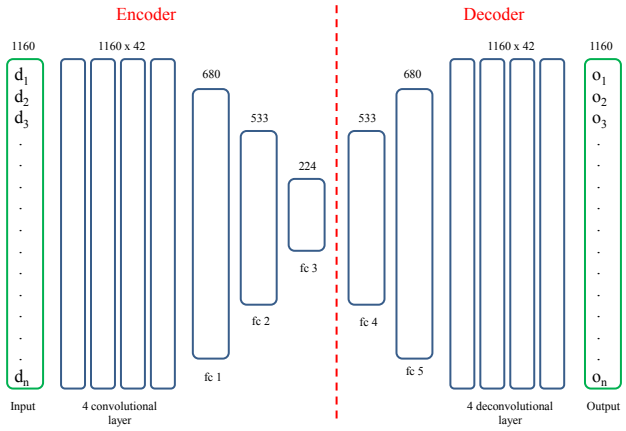


Fig. 2: Structure of the autoencoder used for feature learning from LiDAR scan lines.

an automotive laser scanner on board the vehicle with those generated from an existing point cloud of the environment. A more detailed description of our approach and applied methods is given in the remainder of this paper, which is organized as follows. Section II gives an overview of the presented method. In section III the used autoencoder is presented, section IV describes our localization approach in detail. The experiments and obtained results are presented and discussed in section V. Finally, in section VI conclusions are drawn.

II. OVERVIEW OF THE METHOD

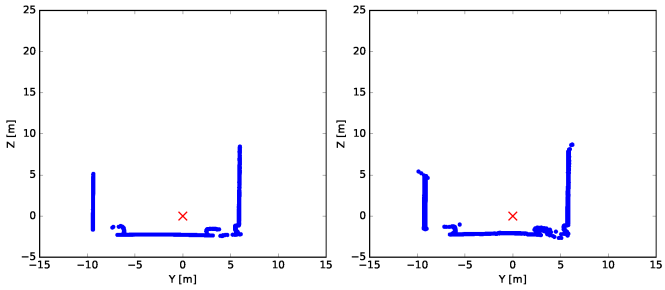
The proposed localization approach is based on matching vertical scan lines captured by a 2D laser scanner mounted on the vehicle with scan lines generated from a reference point cloud of the environment, here called map. The map contains absolute location information which we use to localize the vehicle. It is measured by a vehicle with a highly accurate localization unit, e.g. by using differential GNSS. Fig. 1 shows a flowchart of our approach. An artificial neural network in the form of an autoencoder is trained to obtain significant features from these single scan lines. This step is done for the reference data as well as for the current sequence of scans. The resulting features are assigned to the scan lines and used as descriptive feature vectors for a point-wise matching. Due to the fact that these consecutive scan lines are time series data, which can also be represented as sequence data, we are able to apply existing mining methods from the sequence analysis research field. Using the whole feature vector of every scan line in the sequence analysis would be very time consuming. Also we are aware of the many possible disruptive factors, which influence the shape and therefore also the matching of the scan lines, e.g. lateral deviations between the actual and reference trajectory or dynamic objects on the side. Thus, instead of comparing the feature vectors directly, we first apply a k-means clustering [17], trained by the reference data set, to handle the disruptive factors by determining similar feature vectors and label every single scan line. Afterwards, we are able to match sequences of cluster IDs in order to find exact and similar sequences. To localize our vehicle finally, we choose the position in the map with the highest consensus.

III. AUTOENCODER TO GENERATE FEATURES IN LIDAR DATA

Humans can localize by recognizing the visual appearance of a place in a city. We want to transform this ability to a computer and localize the vehicle on a road in longitudinal direction. Therefore we use LiDAR data of a 2D laser scanner measuring the vertical shape of a street and its surroundings. A typical vertical scan line contains a number of range measurements as shown in Fig. 3a. If we drive on a road for several times, the trajectory may change in latitudinal direction. As a result, the single range measurements also change but the general shape of the scans does not. However the scan lines will not have an exact match because of the inevitable changes in the environment (e.g. caused by dynamic objects like cars) between the times the reference and the test scans are captured. This is why we use a feature vector derived by a convolutional autoencoder to describe the scan and afterwards to localize the vehicle, without taking the range measurements directly into account. Autoencoders were first introduced by Rumelhart et al. [18] in 1985. An autoencoder consists of an encoder and a decoder. The encoder generates a sparse representation of the data, whereby the decoder tries to reconstruct the input so that in the end the input equals the output.

We use TensorFlow [19] to implement the autoencoder. The structure of our network is outlined in Fig. 2. Here the network consists of 13 hidden layers, the encoder with seven hidden layers and the decoder with six hidden layers. Hidden layers 1-4 are 1D convolutions with a stride and padding size of 1, followed by three fully connected layers with a specific number of neurons, for example 680 for the first, 533 for the second and 224 for the third layer. In the evaluation step (section V) we analyze the effect of the number of fully connected layers and neurons to investigate its influence on the performance of the localization method.

The decoder starts with two fully connected layers whose



(a) Input scan. (b) Reconstructed scan.

Fig. 3: An example 2D scan line and its reconstruction by the autoencoder. For visualization the vertical scan lines are transformed to cartesian coordinates. The input data of the autoencoder contains a vector of range measurements ordered by the respective measurement angles.

size equals the size of the first two fully connected layers in the encoder followed by four deconvolutions and finally the output vector. The output is then compared to the input vector to perform the gradient descent optimization using the RMSprop learning rate method [20]. As an activation function we use a rectified linear unit (ReLU).
Fig. 3b shows a reconstructed laser scan of the Mobile Mapping data for a network with a configuration described above. The corresponding input scan is shown in Fig. 3a. It can be seen, that even though the network reduces the dimension from 1160 to 224 values according to the size of the last layer of the encoder, the structure of the scan can still be reconstructed pretty well. In the next step we use the activations of the encoder (here: $n_{feat} = 224$) as a feature vector that describes a single scan line.
By transforming the scan lines to a sparse feature representation we can also reduce the required space to store and transfer the data. This may for example be useful if the reference map on a central server should be updated regularly. In addition, the feature representation could also have positive influence on the computation time. We analyze the run time for a diverse number of features in section V.

IV. LOCALIZATION BY SEQUENCE ANALYSIS

In this step the absolute position of a vehicle is determined by matching its current sequence of scan lines, more specifically its last part of a certain length, to the reference data set. The position of the last item of the segment with the highest consensus corresponds to the actual position of the vehicle.
In order to do the previously mentioned matching, we create the test sequence by using the scan lines of the last n_{ts} time steps. Since this work is based on time series data, to which trajectories can also be counted as they are time series of object locations, we are able to apply existing methods from the research field 'sequence analysis' to our data. Therefore, the segment is transformed into a sequence of items, which are described by the feature vectors provided by the autoencoder (section III). Afterwards, it is matched

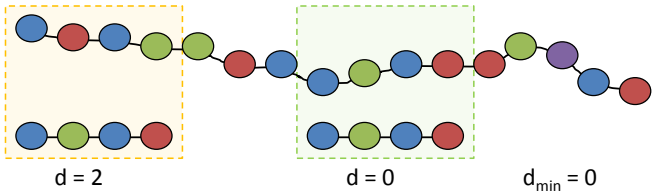


Fig. 4: Every feature vector of the reference map (top) and the current sequence (bottom) is labeled into k clusters, here colored blue, red, green and purple. A sliding window approach combined with an item-wised distance calculation is then used to find the best match between the sequence and the reference map.

to the reference sequences, which have been obtained by transforming the reference data the same way. Instead of comparing the whole feature vector of every sequence item to the reference map, we apply a k-means algorithm to cluster the items based on their feature vectors into k clusters. Using all features for matching is computationally expensive and could not be done in real time, especially for a large reference map. In contrast, comparing two label sequences and counting the number of matches is much faster. Also the storage space can be reduced significantly. We only need to store one label for every position along the trajectory, which can be represented by an integer of 2 Bytes. For a city like Berlin with a street network of about 10,800 km in both driving directions, this results in a storage amount of less then 100 MB.
The clustering parameter k further controls the degree of similarity, which indicates if two items are assigned to the same cluster. A high k means that matched sequences are very similar to each other, although the total number of matching results is low.
However, due to inaccuracies in the data and different measurement conditions between reference and test data (compare section V-A), the probability of finding exact matches, even after having applied a clustering to shrink the alphabet, is still quite low. Accordingly, we calculate a distance between both sequences to determine their consensus instead of checking for item-wise equality. This allows slight deviations between the sequences and means, the lower distance the higher the consensus is. As distance measure we use the Hamming Distance, which counts the number of non-equal item pairs in sequences of equal length. Fig. 4 shows the procedure for matching the reference and test sequences. In the following experiments (section V) we discuss the influence of n_{feat}, n_{ts} and k on the accuracy and robustness of the localization results.

V. EXPERIMENTS AND DISCUSSION

A. Data

We tested our algorithm on LiDAR data derived from a highly accurate and dense Mobile Mapping system.
The data consists of several city parts in Hanover, Germany, with living areas, the city center and also large roads with

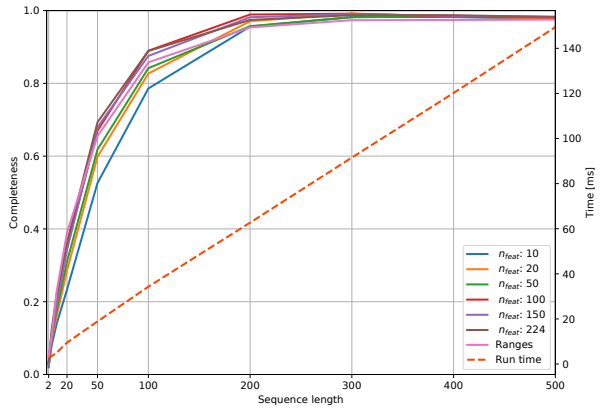


Fig. 5: Completeness and run time for a varied number of features n_{feat} and a sequence length n_{ts} from 2 to 500. The number of clusters k was set to 200.

less structure on the side. We use this data as a reference for our experiments. The overall length of the trajectories is 26 km.

The data was recorded in winter 2015 and spring 2017. For the test data set, however since we did not have the possibility to attach an automotive line scanner to a vehicle, we created the vertical scan lines artificially from a highly dense and accurate real Mobile Mapping data set. For this reason, the scan lines were generated by sampling from a 3D point cloud, gathered by a Riegl VMX-250 Mobile Mapping System [21]. The system contains two laser scanners and a localization unit. The localization is provided by a highly accurate GNSS/INS system combined with an external Distance Measurement Unit. Reference data from the Satellite Positioning Service SAPOS is used to improve the localization, resulting in an accuracy of about 10 to 30 centimeters in height and 20 cm in position in urban areas. The measurement range of the laser scanners is limited to 200 meters, the ranging accuracy is ten millimeters.

Based on this data and on the corresponding trajectories we created vertical 2D line scans. Fig. 3a shows an example for the laser scan, transformed to a 2D representation. The configuration of the scans is comparable to typical automotive scanners: a maximum range of 40 meters, an angular resolution of 0.25 degrees with an opening angle of 290 degrees. The resulting laser scan (one scan line) therefore consists of 1160 range measurements. Please note that we did not add additional noise to the LiDAR data. We set the scanning frequency to 50 Hz and the driving velocity to 15 m/s, yielding a distance of 30 cm between the scan lines in driving direction.

In our experiments we chose a trajectory of 11 km length that extends from Hanover Nordstadt to the city center, see Fig. 7. In contrast to our reference data the test data was recorded in summer (2015), with substantial more vegetation and several more changes in the LiDAR data. We performed the localization for about 35,000 positions along the test trajectory, without any use of GNSS signals.

TABLE I: Effect of the parameters of the neural network with a sequence length n_{ts} of 200 scans and a cluster size k of 200 clusters on the localization accuracy and completeness.

fcl	n_{feat}	kernel	filters	c [%]	RMSE [m]
1	94	2	9	95.7	1.37
1	519	1	37	94.9	1.45
2	413	5	54	95.6	1.39
2	727	4	20	94.7	1.39
2	783	6	4	93.6	1.40
3	**224**	**1**	**42**	**97.4**	**1.37**
3	384	5	59	96.3	1.38
3	511	3	30	97.2	1.39

B. RESULTS

We evaluated the localization performance using two measures: accuracy defined as the root mean square error ($RMSE$) of the resulting positions to the reference coordinates and completeness c defined as the percentage number of correct positions. We consider a position to be correct if the Euclidean distance to the (known) reference position is less than five meters.

First we studied the effect of the parameters of the autoencoder, namely the number of hidden fully connected layers (fcl), the number of neurons in the last layer (which is equal to our feature vector) (n_{feat}) and the kernel size and number of filters in the convolution steps using latin hypercube sampling [22]. Table I shows the the percentage number of successful location determinations (completeness c) on our test trajectory and the corresponding accuracy for several configurations. The sequence length n_{ts} was set to 200 scans, the cluster size k to 200 clusters. It can be seen that the best results are achieved for a configuration with three fully connected layers, 224 neurons, a kernel size of 1 and 42 filters. It is remarkable that a convolution with a kernel size of 1 yields to the best results, as this does not result in a classical convolution. However one by one convolutions (in 2D) are used in some network architectures and were introduced by Lin et al. 2014 [23]. For this specific configuration the completeness rate c is 97.4% and the $RMSE$ is 1.37 m. Hence we used these parameters as a basis for further experiments, where we analyze the effect of n_{feat}, k and n_{ts}.

Fig. 5 shows the results for a varied number of features and a varied sequence length . In general c increases with a higher n_{feat}. At a sequence length of 200, the completeness for n_{feat} = 100 is 98.9%. At a vehicle velocity of 15 m/s (54 km/h) n_{ts} = 200 corresponds to a travelled distance of 60 m. Starting at n_{ts} = 200 the localization results using a scan representation of n_{feat} = 10 features achieves better results than using the original range scans, which can be better seen in Fig. 6 which shows a magnified part of Fig. 5. Using only ten features instead of the whole range scan also reduces the required amount of temporary storage space and has positive influence on the computation time of the labels. A change in the sequence length n_{ts} has a direct influence

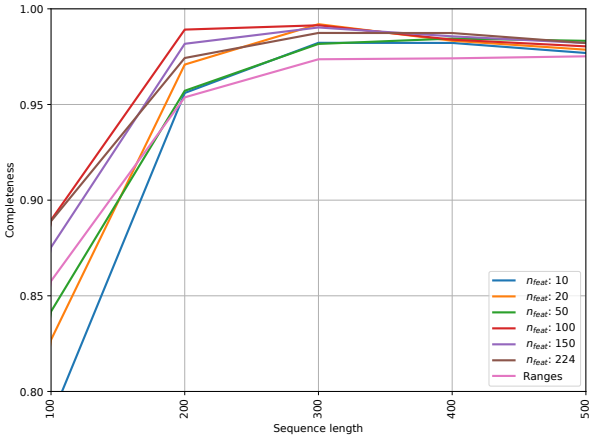


Fig. 6: Completeness for a varied number of features n_{feat} and a sequence length n_{ts} from 100 to 500. The number of clusters k was set to 200.

on the run time of the sequence analysis (see Fig 5). The less labels we have to compare, the faster the algorithm is. In addition, a lower sequence length means that the vehicle needs less measurements and therefore less time to determine its position. At n_{ts} = 200 the computation time for a single global position determination is 62 ms on a 2.6 GHz CPU using only one core. The run time for computing 224 features using the neural network is less than one millisecond. In case of larger reference maps, the computation time could be reduced by parallel processing.

Next to n_{ts} we also varied the number of clusters k in the k-means clustering. Starting at k = 20 the algorithm already performs well, with c = 96.3%. A maximum is reached at k = 50. As the influence of k on the computation time is very low, it is anyway advisable to set k to a high value.

Fig. 7 shows a map of positions, where the localization method achieves the correct results (green), wrong results (red) and where no reference data is available (blue). In area 4 and 5 the localization doesn't work at small town squares. In area 3 the reason of wrong localization results may be caused by the street design, with three to four lanes on the road and low structure at the sides. We assume that additionally using intensity data of the laser scans will reduce these errors. The problems in area 1, 2 and 6 appear because of construction sites or other large changes in the data sets, so that scans of the same position are assigned to different labels. This problem may be solved by a change detection analysis of the data. All these relatively small gaps could also easily be bridged by adding data of an inertial measurement unit in a filter approach.

To show that our method also works on real data, we tested it on reference and test data gathered by an automotive laser scanner of the Oxford RobotCar Dataset [25]. Again using features gathered by a neural network leaded to better results than using the original range measurements. In this case, using a sequence length of $n_{ts} = 200$ yielded a completeness rate of $c = 100\%$ along a 3 km trajectory (see fig. 8). Due to the lack of a highly accurate reference trajectory in the

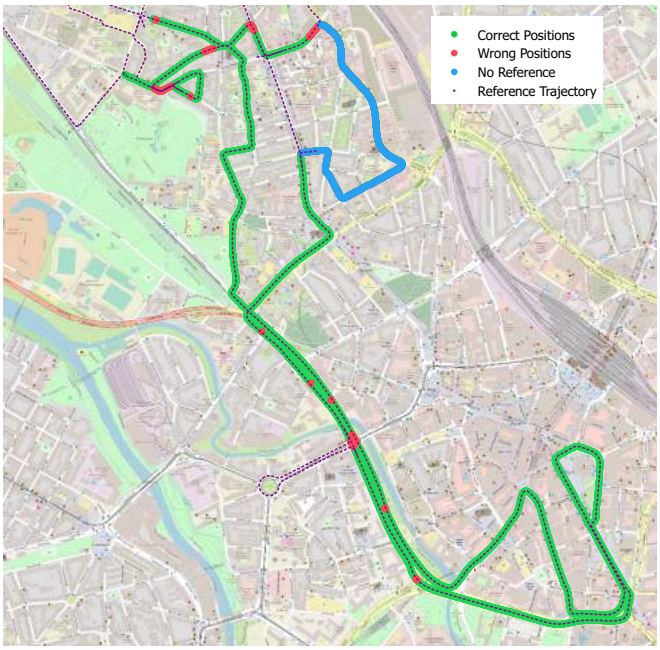


Fig. 7: Visualization of correct (green) and wrong (red) localization results overlaid on an open street map [24] of the study area. Positions where no reference data is available are colored blue. The dashed line represents the reference trajectory. In this case n_{feat} was set to 224, k was set to 200, and n_{ts} was set to 200.

dataset we can not provide information about the accuracy of our method.

VI. CONCLUSION AND OUTLOOK

In this paper we presented an approach to localize a vehicle in urban environments using LiDAR data without any prior information. First we trained a neural network to reduce the feature space of a single laser scan. The network then was used to generate features for every vehicle position along a reference and a test trajectory. The reference trajectory was

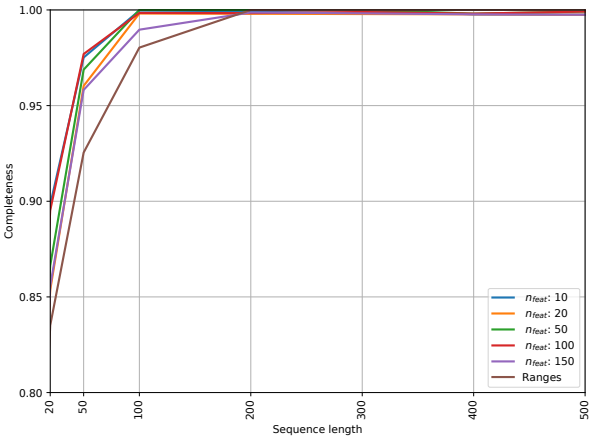


Fig. 8: Completeness for a varied number of features n_{feat} and a sequence length n_{ts} from 20 to 500 for the Oxford RobotCar Dataset. The number of clusters k was set to 200.

used to train a k-means clustering, whose labels build the reference graph. The test trajectory was then clustered by the same k-means. The global localization was done by matching the last n_{ts} labels at a certain test position to the reference graph.

The results show that the method yields up to 99% correct position estimates and a RMS error of about 1.4 m for data derived by measurements of a Mobile Mapping system. The completeness and accuracy decreases with a smaller sequence length and number of clusters in the k-means. The influence of the length of the feature vector is insignificant. The algorithm even performs fine if the laser scan is reduced from 1160 range measurements to ten feature values. Most of the problems in the localization step are caused by changes in the data and low structured street surroundings. We also tested our algorithm on the Oxford RobotCar Dataset, which yielded even better results.

As already mentioned, a change detection algorithm would help to reduce the number of errors. If an area is measured frequently, the reference map should be updated using this data. The data might be measured by a vehicle fleet with a highly accurate localization unit. It would be helpful to store several occurring scans and in our case labels for the same place, like presented in [10].

In future works we also plan to use the intensity data of the reflected laser scans. Thus we can also consider not only the shape of an object, like a building facade, but also the surface. Further, lane markers on the road can help in poorly structured areas to improve the recognition rate.

We created a first reference map for a 26 km trajectory in Hanover, Germany. At the moment the reference data is stored consecutively in the graph, without taking junctions into account. In future the algorithm should work independently of the turning direction. Therefore we have to define a graph structure that also considers the possible turnings while matching the last n_{ts} values of a sequence. To improve the accuracy and to bridge gaps, we also intend to combine the global LiDAR localization with inertial data in a filter approach. It is also planned to extend our method to an even larger data set, e. g. containing the main city parts of Hanover.

ACKKNOWLEDGMENT

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research.

REFERENCES

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

[2] T. Nothdurft, P. Hecker, S. Ohl, F. Saust, M. Maurer, A. Reschka, and J. R. Böhmer, "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.

[3] J. Ziegler, H. Lategahn, M. Schreiber, C. G. Keller, C. Knoppel, J. Hipp, M. Haueis, and C. Stiller, "Video based localization for bertha," in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*. IEEE, 2014, pp. 1231–1238.

[4] 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.

[5] A. Schlichting and C. Brenner, "Localization using automotive laser scanners and local pattern matching," in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*. IEEE, 2014, pp. 414–419.

[6] K. Khoshelham and M. Ramezani, "Vehicle positioning in the absence of gnss signals: Potential of visual-inertial odometry," 2017.

[7] K. Ramezani M., Khoshelham and L. Kneip, "Omnidirectional visual-inertial odometry using multi-state constraint kalman filter," in *International Conference on Intelligent Robots and Systems (IROS17)*. IEEE, 2017.

[8] TomTom N.V., "RoadDNA - Robust and scalable localization technology," TomTom N.V., Tech. Rep., 2017. [Online]. Available: https://automotive.tomtom.com/

[9] L. Li, M. Yang, C. Wang, and B. Wang, "Road dna based localization for autonomous vehicles," in *Intelligent Vehicles Symposium (IV), 2016 IEEE*. IEEE, 2016, pp. 883–888.

[10] W. Maddern, G. Pascoe, and P. Newman, "Leveraging experience for large-scale lidar localisation in changing cities," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1684–1691.

[11] H. Hu and N. Wei, "A study of gps jamming and anti-jamming," in *Power Electronics and Intelligent Transportation System (PEITS), 2009 2nd International Conference on*, vol. 1. IEEE, 2009, pp. 388–391.

[12] J. Coffed, "The threat of gps jamming: The risk to an information utility," *Report of EXELIS, Jan*, 2014.

[13] A. J. Kerns, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, "Unmanned aircraft capture and control via gps spoofing," *Journal of Field Robotics*, vol. 31, no. 4, pp. 617–636, 2014.

[14] M. Bosse and J. Roberts, "Histogram matching and global initialization for laser-only slam in large unstructured environments," in *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, 2007, pp. 4820–4826.

[15] G. D. Tipaldi, L. Spinello, and W. Burgard, "Geometrical flirt phrases for large scale place recognition in 2d range data," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 2693–2698.

[16] M. Himstedt, J. Frost, S. Hellbach, H.-J. Böhme, and E. Maehle, "Large scale place recognition in 2d lidar scans using geometrical landmark relations," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 5030–5035.

[17] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[18] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," DTIC Document, Tech. Rep., 1985.

[19] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: http://tensorflow.org/

[20] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, 2012.

[21] RIEGL Laser Measurement Systems GmbH, "RIEGL VMX-250," RIEGL Laser Measurement Systems GmbH, Tech. Rep., 2012. [Online]. Available: http://www.riegl.com

[22] M. Stein, "Large sample properties of simulations using latin hypercube sampling," *Technometrics*, vol. 29, no. 2, pp. 143–151, 1987.

[23] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.

[24] OpenStreetMap contributors, "Planet dump retrieved from https://planet.osm.org ," https://www.openstreetmap.org , 2017.

[25] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 Year, 1000km: The Oxford RobotCar Dataset," *The International Journal of Robotics Research (IJRR)*, vol. 36, no. 1, pp. 3–15, 2017.