2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

Reliable Data Association for Feature-Based Vehicle Localization using Geometric Hashing Methods

Isabell Hofstetter, Michael Sprunk, Florian Ries, Martin Haueis

***Abstract*— Reliable data association represents a main challenge of feature-based vehicle localization and is the key to integrity of localization. Independent of the type of features used, incorrect associations between detected and mapped features will provide erroneous position estimates. Only if the uniqueness of a local environment is represented by the features that are stored in the map, the reliability of localization is enhanced.**

In this work, a new approach based on Geometric Hashing is introduced to the field of data association for feature-based vehicle localization. Without any information on a prior position, the proposed method allows to efficiently search large map regions for plausible feature associations. Therefore, odometry and GNSS-based inputs can be neglected, which reduces the risk of error propagation and enables safe localization.

The approach is demonstrated on approximately 10min of data recorded in an urban scenario. Cylindrical objects without distinctive descriptors, which were extracted from LiDAR data, serve as localization features. Experimental results both demonstrate the feasibility as well as limitations of the approach.

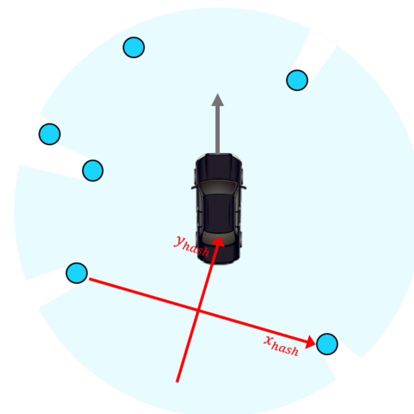


Fig. 1: The concept of Geometric Hashing for localization: Introduction of additional coordinate frames, here depicted as the *hash frame* in red, to efficiently associate the features using a hash table that was previously computed for a feature-based map.

I. INTRODUCTION

Feature-based vehicle localization approaches have been studied thoroughly in the past years and their practical application is known to have great value. These methods depend on a previously generated map that contains various features, also called *landmarks*. In order to localize the vehicle in map coordinates, extracted features from the vehicle's surrounding are associated with the map data. These association constraints pose an optimization problem, which is solved to find an accurate pose estimation.

Available localization methods differ mostly in the kind of features that are used. There are methods that make use of very dense information such as visual point features which are extracted from camera images [1], or point clouds that are obtained from LiDAR or RADAR measurements [2]. Other approaches that focus on a reduced map size and reusability of features, suggest the usage of features with a semantic meaning such as road markings [3], [4], or geometric primitives [5].

The key challenge which all these methods face, is correct and reliable feature association. In order to provide safe localization, a proof of the probability of incorrect associations being sufficiently low is desirable. Most methods do not have a mechanism to do so and rely on a temporal component, e.g. a previous pose estimate, to argue that no other association is feasible. This introduces a certain risk for error propagation and, therefore, also safe localization.

This contribution introduces the concept of Geometric Hashing to the field of data association for feature-based vehicle localization. Geometric Hashing was initially introduced for object recognition purposes and is a fast method to search large data sets for a given sample of features [6], [7]. This work focuses on the adaption of the method for vehicle localization purposes and explores and demonstrates its feasibility as well as limitations. The presented approach provides the means for data association without any assumptions or knowledge on prior positions. It allows for feature association based only on the geometric information provided by the extracted features while evaluating all possible hypotheses. This eliminates the risk of error propagation and enables safe localization. The method can be used as a safety localization which complements a state-of-the-art solution and provides the means for the online validation of feature associations.

In the following, an overview of related work is given and the Geometric Hashing concept as it was proposed by Lamdan et al. in [6] is briefly recapitulated. Then, an association algorithm that utilizes Geometric Hashing is introduced for the online data association for vehicle localization. Finally, experimental results verify the potential of the proposed approach and explore its limitations. We derive and test the method on a dataset representing an urban loop of $3.8\,\mathrm{km}$ in Sindelfingen, Germany, where cylindrical structures, such as poles, signs, and trees, are used for localization. The method,

All authors are with the Daimler AG Group Research & MBC Development, Sindelfingen, Germany. {isabell.hofstetter, michael.sprunk, florian.ries, martin.haueis}@daimler.com

978-1-7281-7395-5/20/$31.00 ©2020 IEEE

however, is not limited to this specific kind of landmarks and can easily be adapted to arbitrary localization features.

II. RELATED WORK

Geometric Hashing has been applied in many different fields of research to various use cases as for example in computer vision for object recognition [8], [9] or structural alignment of proteins [10].

In the broad field of feature-based vehicle localization, however, Geometric Hashing was barely touched. In [11], the authors utilize Geometric Hashing for visual localization in large city environments from aerial images. They propose to use buildings as features that are extracted from satellite image data. A different hashing based approach was suggested in [12], where the authors propose to use scan points itself in order to avoid the feature extraction step and apply Geometric Hashing to entire scans of the environment.

These works either focus on finding a very rough global estimate of the vehicle's position or the use of dense feature data. The latter is also the case in visual place recognition, which is the solution to the problem of determining if an image can be associated to a place that has been seen before [13], [14]. These methods depend on entire images that are searched for correspondences. However, for vehicle localization this is usually only used for initialization.

In contrast, this work focuses on the use of sparse features while still trying to find a reasonably accurate position in the map frame. We show the potential of the approach, not as a stand-alone method for data association, but as a second mainstay to validate and confirm data association online.

To the author's knowledge, this is something that has not been studied before. Most localization algorithms that try to provide safe localization depend on prior positions to find reliable feature associations [5], [15]. However, time dependent information always introduces a certain risk of error propagation, which this contribution tries to eliminate.

III. GEOMETRIC HASHING

Geometric Hashing was originally introduced by Lamdan et al. in the field of computer vision for object recognition purposes [6]. It is a well known technique for matching geometric features that have undergone a transformation. In an offline step, features of interest are represented in a variety of coordinate systems that are defined by the features themselves and stored in a quickly searchable, tabular format. Online, this data base is searched for the best match.

In the following sections, the concept of Geometric Hashing is briefly revised and the two phases, training phase & recognition phase, are discussed separately. For more detail, the authors refer to [6], [7] or [16].

A. Training phase

During an offline phase, called the *preprocessing* or *training phase*, a hash table is generated for the objects of interest or *models*. Let M be a list of models for recognition. For each model $m_i \in M$, $i = 1, \ldots, N$, all feature points S_i with $|S_i| = N_i$ are extracted. Each ordered pair of point features (p, q) with $p, q \in S_i$ and $p \neq q$ represents a *basis* or *basis pair*. Let K_i be the set of basis pairs of model m_i. The cardinality of this basis set is $|K_i| = N_i(N_i - 1)$. Each basis $k = (p, q) \in K_i$ defines a local coordinate frame $O_{(p,q)}$, called the *basis frame* or *geometric basis*. The origin of this frame is defined by the midpoint between the two feature points p and q. The x-axis is represented by the axis passing through p and q. For each basis $k \in K_i$, the remaining feature points $S_i \setminus k$ are transformed into the corresponding basis frame and quantized appropriately. Finally, a hash function is applied to each feature point and the hashes are stored in a hash table H.

B. Recognition phase

In an online step, a given input sample S can be searched for the objects of interest with the help of the hash table. This phase is called the *recognition phase*. Here, an arbitrary ordered basis pair k of two random feature points $p_1, p_2 \in S$ is generated. This defines a geometric basis as described before. All remaining points of interest $S \setminus k$ are transformed into this frame, quantized and hashed in the same fashion as during the training phase. Finally, the hash table is searched for matching hash values and a vote is cast for each correspondence between the newly generated basis and each hash table layer. If sufficient feature correspondences are found, the matching model is returned.

C. Noise model

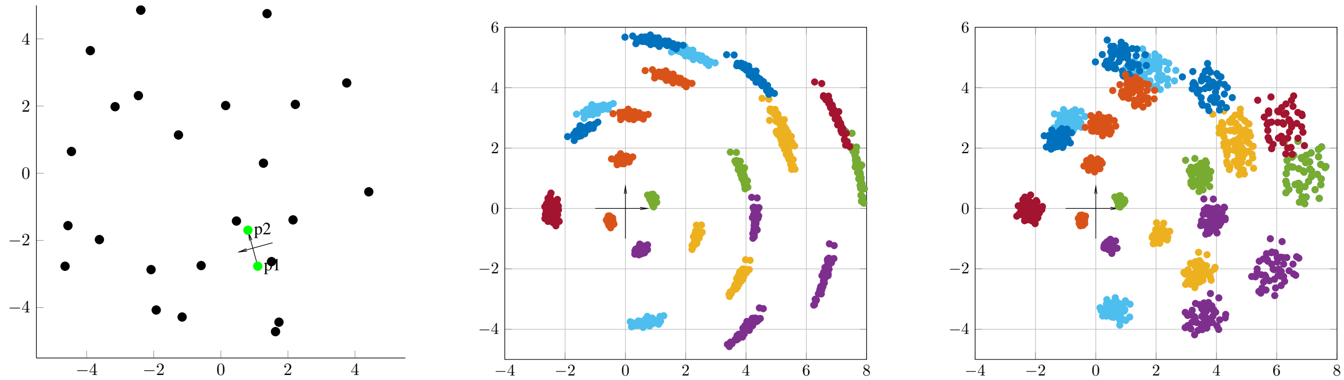
If the given input sample S is affected by noise, the positional error of the computed hash locations in the basis frame has to be described by a valid noise model. This topic was studied in [16]. Rigoutsos shows that the positional noise on the feature points in the basis frame depends heavily on the chosen basis pair, the position of the point relative to the basis origin as well as the kind of transformation used. In his work, he derives a noise model for similarity as well as affine transformations.

In the following, the main results for similarity transformations will be briefly recapitulated and a noise model for rigid transformations is proposed. For the sensor noise on each single feature point, we will assume a Gaussian distribution with standard deviation σ, which is centered at the "true" location of the feature.

1) Similarity Transformation: If the models of interest are allowed to undergo rotation, translation, and scaling, a similarity transformation can be used for hashing. In this case, given p_1 and p_2 as a basis pair, every other feature p can be represented in the basis frame defined by p_1 and p_2 by

$$p = p_0 + (p_2 - p_1) \cdot u_s + (p_2 - p_1)^{\perp} \cdot v_s,$$

where p_0 represents the origin and (u_s, v_s) the coordinates of p in the basis frame. For this kind of transformation, the resulting noise on the features in the basis frame can be approximated very well by a Gaussian distribution with



(a) Random sample features, where features p_1 and p_2, depicted in green, were chosen as a basis pair.

(b) Simulated noisy measurements transformed by a rigid transformation into the basis frame defined by p_1 and p_2.

(c) Simulated noisy measurements transformed by a similarity transformation into the basis frame defined by p_1 and p_2.

Fig. 2: Noise simulation for rigid and similarity transformations.

standard deviation

$$\Sigma_s = \frac{(4||(u_s, v_s)||^2 + 3)\sigma^2}{2||p_2 - p_1||^2} \tag{1}$$

as was shown in [16].

2) Rigid Transformation: If the models of interest are allowed to undergo rotation and translation but no scaling, rigid transformations are considered for hashing. Again, given a basis pair consisting of p_1 and p_2, every other feature p can be represented in the basis frame by

$$p - p_0 = \frac{(p_2 - p_1)}{||p_2 - p_1||} \cdot u_r + \frac{(p_2 - p_1)^\perp}{||p_2 - p_1||} \cdot v_r,$$

where p_0 represents the origin and (u_r, v_r) the coordinates of p in the basis frame.

In this case, the noise on the feature positions in the basis frame cannot be described accurately by a Gaussian distribution, which is reflected by the simulation results in Fig. 2. Here, a random sample of feature points was generated, which is visualized in Fig. 2a. Noisy measurements were simulated and transformed into the basis frame defined by p_1 and p_2 using a rigid transformation (see Fig. 2b) and a similarity transform (see Fig. 2c), respectively. Depending on the chosen basis pair and the feature locations with respect to the basis frame, the noise in the angular component is enhanced compared to the noise in radial direction.

In the following, an over-bounding of the noise in radial direction is suggested in order to obtain a Gaussian distribution. This results in a standard deviation of

$$\Sigma_r = \frac{(4||(u_s, v_s)||^2 + 3)\sigma^2}{2||p_2 - p_1||}, \tag{2}$$

which equals the standard deviation Σ_s in equation (1) scaled by the basis pair distance $||p_2 - p_1||$.

IV. LOCALIZATION USING GEOMETRIC HASHING

For the purpose of data association for localization, a modified Geometric Hashing method is derived in the following subsections. This suggested method focuses on hashing using rigid transformations since scale relations provide useful information for localization and should therefore be preserved. A similar hashing approach for the offline processing of feature-based maps was also derived in [17].

A. Generation of the Hash Table

For a given feature-based map $M = \{f_1, f_2, \ldots, f_n\}$, where n is the number of map features, a hash table is generated during an offline process. This hash table is not created to contain information about certain models, but instead should hold sufficient information about each local environment in the map. Therefore, the basis set K is chosen as follows: Basis pairs $k = (f_i, f_j)$ with $f_i, f_j \in M$ for $i, j \in \{1, \ldots, n\}, i \neq j$ are generated if and only if

$$d = ||f_i - f_j|| < d_{basis}$$

for a given maximum basis pair limit d_{basis}. This way only bases that are representative for a local environment in the map, are generated, which reduces the size of the resulting hash table while still ensuring a more than sufficient coverage of the whole map.

Once all desired basis pairs are found, for each basis $k \in K$ the origin $z(k)$ of the corresponding basis frame is computed and the features of interest I_k are determined. Features of interest are all features in a local surrounding of the basis origin $z(k)$, which is limited by the inclusion radius r_{incl}. Then, the coordinates of the set of features I_k in the basis frame are calculated using a rigid transformation. Fig. 3 illustrates the main parameters used in this process.

Finally, the features of interest I_k are quantized and hashed according to a given bin size s_{bin}. In addition to the hash values, information about the noise distribution for features in the basis frame is stored in the hash table according to equation (2). Each hash table layer now holds information about a local map environment in a coordinate frame defined by two arbitrary map features.

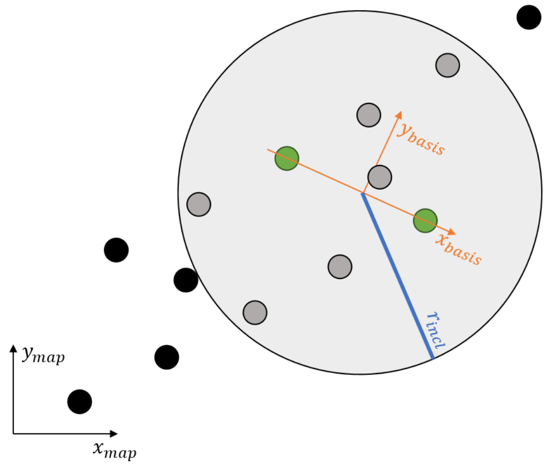


Fig. 3: Illustration of the main symbols and variables involved in the hash table generation for a given map section: Basis pair k (green), basis frame (orange), inclusion radius r_{incl} (blue), features of interest I_k (gray).

B. Feature Association

Given a sample of detected features D and the previously generated hash table H, feature associations can now be determined by searching the hash table. This searching and validation process is compactly summarized in Algorithm 1.

First, two arbitrary detections $p_1, p_2 \in D$ are chosen, which define a geometric basis $O_{(p_1,p_2)}$ just as before. All feature detections D are then transformed by a rigid transformation into this basis frame, quantized, and hashed accordingly. For each computed hash value, appropriate bins in the hash table H are accessed and a vote is given to each basis found there. In this step, not only the exact bin is accessed but also its direct neighborhood to ensure that also noisy measurements that fell into neighboring bins can be associated.

Then, the set of bases B is determined that reached a predefined threshold for the number of votes. These are all candidate bases that could contain valid associations. As a secondary check, for each basis $O_k \in B$, the Mahalanobis distance

$$d_{mal} = \sqrt{(p-\mu)^T \Sigma^{-1} (p-\mu)} \tag{3}$$

is computed for each sample measurement $p \in O_{(p_1,p_2)}$ and $f \in O_k$ with $f \sim \mathcal{N}(\mu, \Sigma)$. The information on the distribution was previously added to the hash table during the generation process. A weight w based on the Mahalanobis distance is then stored for each candidate association. Finally, the previously described steps can be repeated for each available basis pair $k = (p_i, p_j)$ with $p_i, p_j \in D$, $i \neq j$ for validation purposes. In a final step, the weights are summed for each candidate association and the final associations are chosen accordingly.

Algorithm 1 FEATURE ASSOCIATION

Input: Hash Table H, Sample of detected features D.
Output: List of feature associations.

1) Randomly choose a basis pair (p_1, p_2) with $p_1, p_2 \in D$.
2) Compute the coordinates for each $p \in D$ in the basis frame $O_{(p_1,p_2)}$, quantize, and hash each coordinate.
3) Access the appropriate hash table bins in H for each hash and cast a vote for each geometric basis found there.
4) Histogram all bases that got one or more votes during step 3 and determine those bases B that received more votes than a predefined threshold: Each of these bases could contain valid associations.
5) For each basis $O_k \in B$ found in step 4, compute the Mahalanobis distance d_{mal} between features in $O_{(p_1,p_2)}$ and features in O_k.
6) Store a weight w based on the Mahalanobis distance d_{mal} for each candidate association.
7) Repeat steps 1 - 6 for all possible keys $k = (p_i, p_j)$ with $p_i, p_j \in D, i \neq j$.
8) Sum all weights per candidate association and choose associations based on summed weights.

V. EXPERIMENTAL RESULTS

A. Dataset

The dataset used for the demonstration and evaluation of the approach was collected with the vehicle *BerthaONE* [18]. It was recorded in Sindelfingen, Germany, and represents a loop of $3.8\,\mathrm{km}$ in an urban area which results in approximately $10\,\mathrm{min}$ of driving time. The features that serve to demonstrate our approach are cylindrical objects, such as signs, traffic lights, poles, and trees, that were extracted from LiDAR data as described in [5].

The map that was used to demonstrate this approach, contains 738 cylindrical features. All cylinders are represented by the coordinates of their center point on the ground plane in map coordinates (x, y). As a reference solution for validation purposes, we use the approach described in [1]. A visualization of the map and the reference trajectory can be seen in Fig. 4.

B. Hash Table

A hash table was generated for the previously mentioned map. The parameters for the hash table generation were chosen as stated in Table I. Those values were chosen in order to ensure a good coverage of the map and at the same time, to limit the number of basis frames. To capture the true distances between features, a rigid transformation was used to transform the features into the basis frames. The resulting hash table contains 37236 geometric basis frames or hash layers and the serialized data results in a file size of $305.8\,\mathrm{MB}$. This can easily be reduced by carefully selecting bases to be generated, which was not in the scope of this work.

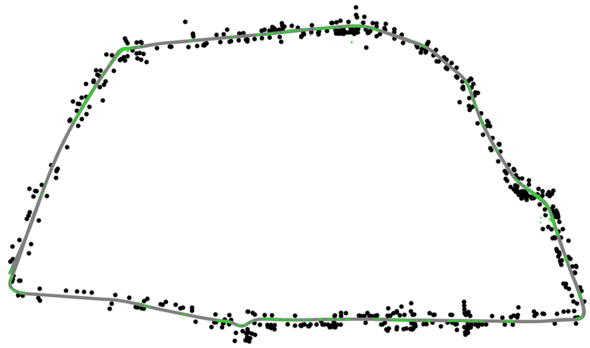


Fig. 4: Feature map containing 738 cylinderical features (black), the reference trajectory used for evaluation (gray), and localization poses where associations were found using the approach described in Algorithm 1 (green).

TABLE I: Hash Table Properties

Property	Variable	Value
Bin size	s_{bin}	0.1 m
Basis Limit	d_{basis}	60 m
Inclusion Radius	r_{incl}	100 m
Transformations		Rigid
Layers		37236
Hash Entries		1603063
File size		305.8 MB

C. Localization Framework

The localization framework used to demonstrate the approach represents a single shot localization system. It is not designed to be utilized independently but instead can be used as a complementary safety localization that provides input on how reliable data association is. The framework is depicted in Fig. 5. The contribution of this work, which was described in the previous sections, is highlighted in gray.

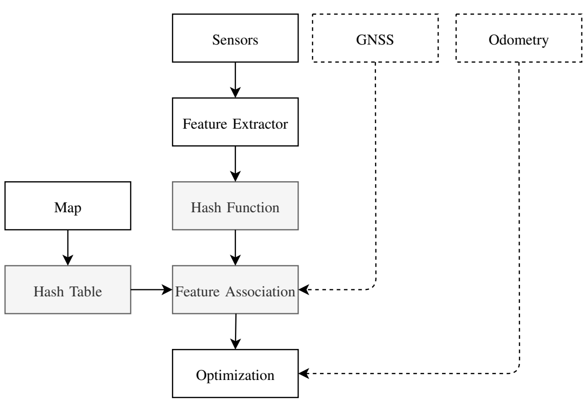


Fig. 5: Overview of the suggested localization framework using Geometric Hashing for data association.

As an input to the feature association, it is assumed to have a map containing point features as well as the corresponding feature extractor that provides the detected features online. The output consists of a list of associations between mapped and detected features. These associations pose constraints for an optimization problem, which is solved for the final localization solution.

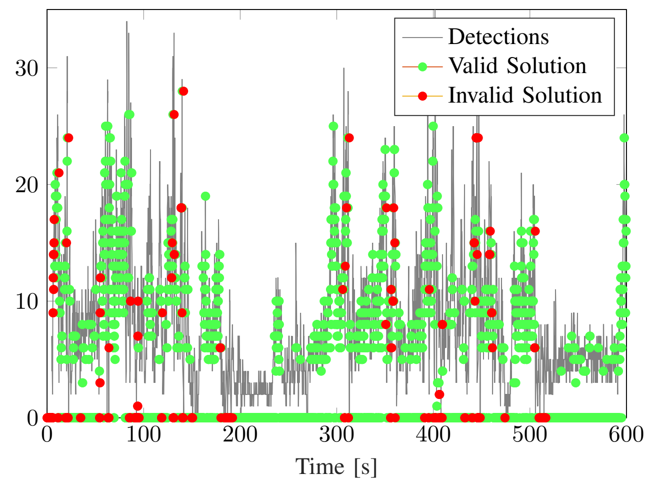


Fig. 6: Number of feature detections and found associations versus time. Color code: Pose estimate classified as valid (green), otherwise (red). For zero associations, the prior pose is propagated using odometry measurements.

A GNSS receiver and an odometry unit are added to the framework in dashed lines as optional components. The biggest advantage of the proposed method lies in the fact that no propagation of prior poses is needed to find valid associations and therefore, we mostly neglect such information. However, the odometry output can be used to provide a continuous solution in cases where the suggested method is unavailable. A GNSS-based solution helps to validate associations by determining an extremely rough approximation of the current area of interest, but is also not necessary in theory.

D. Localization results

The potential of the proposed approach is mainly assessed by the availability of the system, the number of associations given a certain number of detections per time as well as the processing time. Nevertheless, results on the accuracy of the resulting position estimates will also be discussed. Availability plots can be seen in Fig. 4 and 6. Fig. 4 shows the map, the reference trajectory, and poses are highlighted in green where associations were found. Fig. 6 visualizes the number of detections and associations versus time. We classify the result in each time step based on the accuracy of the resulting position estimate. If an accuracy of less than 5 m in longitudinal and lateral direction and a yaw error of less than 30° was achieved, we call it a valid solution. All time steps in Fig. 6 when a valid solution was computed, are represented in green, otherwise in red.

Overall, associations are found in 34.4% of the total time. This is partially due to the number of detections given at each time step. Only in 62% of the time, five or more detections were found. In these cases, an overall availability of approximately 51% is achieved.

In total, 92.8% of all localization solutions are valid. This includes solutions that were propagated due to the fact that no

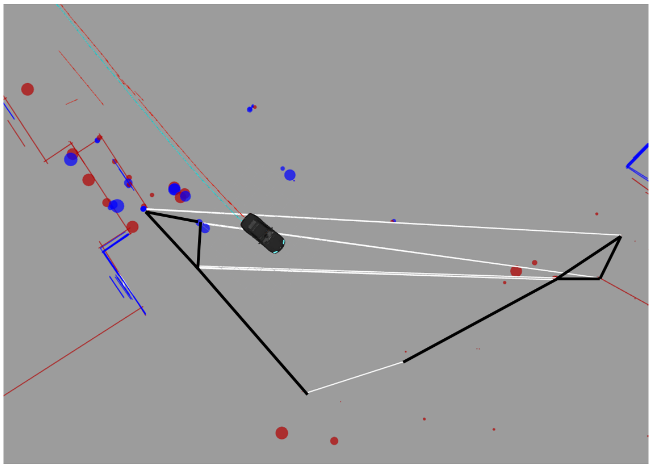


Fig. 7: Ambiguities lead to wrong associations. This snapshot shows an ambiguous feature pattern that was wrongly associated. It is highlighted by the use of black lines connecting the cylinders.

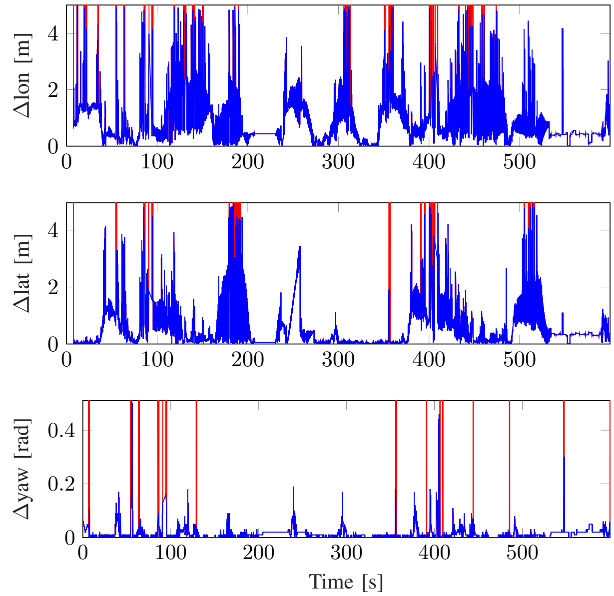


Fig. 9: Localization accuracy in longitudinal and lateral direction as well as orientation. Color code: valid solution (blue), invalid solution (red).

association could be found. In those cases where associations were found, 94.2% of the pose estimates were valid and 5.8% were invalid, respectively. This is due to ambiguous feature patterns in the map. Fig. 7 shows a scenario where a wrong association was chosen due to a similar feature pattern in the map. The chosen associations are connected by white lines, the feature pattern is redrawn by the use of black lines for better visualization.

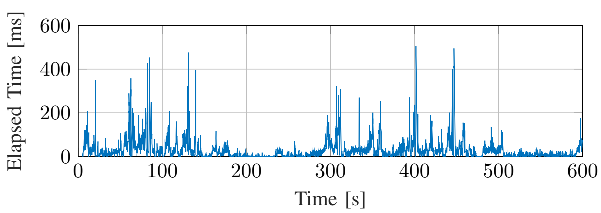


Fig. 8: Elapsed processing time during feature association. The feature association was run on a machine with an Intel Core i7-3770K CPU @ 3.50GHz.

Processing time depends heavily on the number of features N_f that were extracted in each time step (see Fig. 8). This is due to the fact that $N_f(N_f - 1)$ geometric bases are generated per time step. However, this can easily be reduced by limiting the number of online generated basis frames. Also, the method can be very well parallelized which leaves additional room for improvement.

Finally, the localization accuracy was evaluated based on the position error in longitudinal direction Δlon and lateral direction Δlat as well as the yaw angle error Δyaw. The error plots for the evaluation drive are shown in Fig. 9. The overall RMS error amounts to $1.3\,\mathrm{m}$ in longitudinal, $1.03\,\mathrm{m}$ in lateral direction and $0.14°$ in the yaw angle.

VI. CONCLUSIONS AND FUTURE WORK

This work introduces the concept of Geometric Hashing to the field of data association for feature-based vehicle localization and explores both the feasibility as well as limitations of the approach.

A Geometric Hashing based method is derived that enables data association for vehicle localization without any prior position estimate and relies solely on the detected features per time step.

The suggested approach is demonstrated on a data set recorded in an urban area where cylindrical structures, which are represented by their 2D position in the map frame, serve as localization features. A hash table is generated during an offline process for a map containing 738 features and used online for data association.

Experimental results demonstrate the advantages as well as the limitations of the approach: A map covering $3.8\,\mathrm{km}$ of roads can be searched for feature associations in $32\,\mathrm{ms}$ on average in order to determine the vehicle's pose. This is accomplished without any knowledge on the vehicle's prior position which eliminates the risk of error propagation. Additionally, the method can be utilized to detect ambiguities in the map data. However, in the presence of ambiguities, the method might choose the wrong association during the association step.

Future work will focus on an accurate noise model in polar coordinates, a thorough validation of associations, and the use of information about ambiguities in maps. We plan to use this approach as a safety localization which complements a state-of-the-art solution and, therefore, offers the means for the online validation of feature associations.

ACKNOWLEDGMENT

The map and data set was recorded and kindly provided by the Institute of Measurement and Control System of the Karlsruhe Institute of Technology (KIT). The authors would like to thank their colleagues in Karlsruhe for their support.

REFERENCES

[1] M. Sons, M. Lauer, C. G. Keller, and C. Stiller, "Mapping and localization using surround view," in *IEEE Intelligent Vehicles Symposium (IV)*, pp. 1158–1163, 2017.

[2] F. Schuster, C. G. Keller, M. Rapp, M. Haueis, and C. Curio, "Landmark based radar SLAM using graph optimization," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2559–2564, 2016.

[3] M. Schreiber, C. Knöppel, and U. Franke, "Laneloc: Lane marking based localization using highly accurate maps," in *IEEE Intelligent Vehicles Symposium (IV)*, pp. 449–454, 2013.

[4] F. Poggenhans, N. O. Salscheider, and C. Stiller, "Precise Localization in High-Definition Road Maps for Urban Regions," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2167–2174, 2018.

[5] J. Kümmerle, M. Sons, F. Poggenhans, M. Lauer, and C. Stiller, "Accurate and Efficient Self-Localization on Roads using Basic Geometric Primitives," in *2019 IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[6] Y. Lamdan and H. J. Wolfson, "Geometric hashing: A general and efficient model-based recognition scheme," 1988.

[7] H. J. Wolfson and I. Rigoutsos, "Geometric hashing: An overview," *IEEE Computational Science and Engineering*, vol. 4, no. 4, pp. 10–21, 1997.

[8] Y. Lamdan, J. T. Schwartz, and H. J. Wolfson, "Object recognition by affine invariant matching," in *Proceedings CVPR'88: The Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 335–344, IEEE, 1988.

[9] A. C. Roy, K. Hossin, M. P. Uddin, M. A. Al Mamun, M. I. Afjal, and A. M. Nitu, "Detection and classification of geometric shape objects for industrial applications," *Advancement in Image Processing and Pattern Recognition*, vol. 1, no. 2, pp. 11–19, 2018.

[10] N. Leibowitz, Z. Y. Fligelman, R. Nussinov, and H. J. Wolfson, "Multiple structural alignment and core detection by geometric hashing.," in *ISMB*, pp. 169–177, Citeseer, 1999.

[11] T. Senlet, T. El-Gaaly, and A. Elgammal, "Hierarchical semantic hashing: Visual localization from buildings on maps," in *2014 22nd International Conference on Pattern Recognition*, pp. 2990–2995, IEEE, 2014.

[12] M. Tomono, "A scan matching method using euclidean invariant signature for global localization and map building," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 1, pp. 866–871, IEEE, 2004.

[13] S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, "Visual place recognition: A survey," *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 1–19, 2015.

[14] I. Ulrich and I. Nourbakhsh, "Appearance-based place recognition for topological localization," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 2, pp. 1023–1029, Ieee, 2000.

[15] M. Joerger and B. Pervan, "Quantifying Safety of Laser-Based Navigation," *IEEE Transactions on Aerospace and Electronic Systems*, 2018.

[16] I. Rigoutsos, "Massively parallel bayesian object recognition," tech. rep., New York University, NY, Department of Computer Sciences, 1992.

[17] I. Hofstetter, M. Sprunk, F. Schuster, F. Ries, and M. Haueis, "On Ambiguities in Feature-based Vehicle Localization and their A Priori Detection in Maps," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1192–1198, IEEE, 2019.

[18] Ö. Ş. Taş, N. O. Salscheider, F. Poggenhans, S. Wirges, C. Bandera, M. R. Zofka, T. Strauss, J. M. Zöllner, and C. Stiller, "Making bertha cooperate–team annieways entry to the 2016 grand cooperative driving challenge," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 4, pp. 1262–1276, 2017.