Localization for autonomous vehicles using provider maps

E. RAMONA STEFANESCU F. ERLIK NOWRUZI
CONG WU OLEG ZHUKOV
TING YUAN DOMINIK NUSS
Daimler AG
Mercedes-Benz Research & Development North America, Inc.
Sunnyvale, CA 94085, USA

elena_ramona.stefanescu@daimler.com |
fnowruzi@uottawa.ca |
cong.wu@student.kit.edu
oleg.v.zhukov | ting.yuan | dominik.s.nuss
@daimler.com

***Abstract*—In this paper we present a probabilistic approach for an autonomous vehicle to localize itself using landmarks and features. One of the main tasks during localization is data association which can be extremely cumbersome when dealing with poorly calibrated sensors, noisy measurements and failures to detect objects. The availability of highly accurate third party maps is often overlooked in the localization process, in favor of self-created sensor maps. The main challenge in this paper was to transform a dense grid map into a layer that could be overlaid on top of the provider map and associate the features with the landmarks. With measurements correctly associated to the appropriate map landmarks, the vehicle pose is determined using an improved estimation technique. This involves an Extended Kalman Filter pose estimation using a validation gate. This method has the advantage of efficient data fusion from multiple sensor measurements, and the ability to incorporate explicit sensor and process uncertainty models.**

I. INTRODUCTION

The localization problem is a key issue in making autonomous vehicles (AV's). Advances in computing hardware coupled with the availability of different types of sensors have brought the dream of AV closer to reality. However, accurate localization of a vehicle is a challenging task as GPS available on the market is not designed for cm-level accuracy applications.

In the context of automated driving, localization is the function that estimates the pose of the vehicle relative to a map of the environment. Usually the localization process involves five main concepts: a) a map of the surrounding environment, b) a model of the motion of the ego vehicle, c) sensor data from the surrounding environment, d) a method for matching sensory data to the map, and e) an algorithm that computes estimates of the vehicle pose given a correspondence among sensors data, ego motion, and the map [1], [2]. A vast majority of localization approaches are based on learning models of the environment and using them for precise navigation. However, with the increase in the accuracy and availability of third party maps, an integration with range sensor data is important for a better calibration and fitting of these models.

There is an ongoing debate about whether features or landmarks are more suitable for localization. For the purpose of this paper, we defined the following terms:

- *An object* is a complete static item (with boundaries). An object usually makes sense to a human (semantic) and it can be reliably classified by sensors.
- *A landmark* is a static object that can be used for localization purposes. Landmarks have a fixed and known position, which is stored in the map. Examples include curb, guard rails, poles, traffic signs, lane markings, and stop bars. Ideally, landmarks are reliably detected and localized, robust to noise, and invariant to the type of sensors adopted.
- *A feature* is a sensor-specific data set (such as gradient or infrared reflectivity information). A stable feature is a feature that appears at the same exact location over multiple drives. They are provided by the grid map.

Although most existing prototype autonomous vehicles adopt feature-based localization techniques, features have a set of disadvantages [3]. Since features are heavily dependent on the sensors, we have to use, at drive time, the same vehicle configuration that was used to extract the features. This means that different feature sets may be required for different car models. Furthermore, installing the same sensors on the same vehicle but in different positions might also require regenerating the feature set. This is acceptable for a small fleet of research vehicles but is not a long-term solution for a bigger fleet.

A further concern with features is their stability. Features are not as stable as landmarks and suffer from variations in light, time of day, weather conditions, and a number of other factors.

A concern with landmarks is their availability and density. Given a priori knowledge of the landmark position, a vehicle can detect the landmark and determine its relative distance to calculate its own position. In practice, however, landmarks can be highly sparse – requiring a vehicle to navigate solely with the use of features until a new landmark can be detected.

A. Previous work

During the past years, significant progress has been made towards the development of localization algorithms, as well Simultaneous Localization and Mapping (SLAM). However, little progress has been shown in the area of integrating high quality and high definition provider map data with self-made sensor maps.

Most of the localization methods use a variation of the Kalman filter [4], [5] or a particle filter [6], [7], [8] to estimate the vehicle pose. Another distinction can be made based on the sensors that are utilized besides an Inertial Measurement Unit (IMU) or a GPS receiver. Because of its relatively high resolution and precision, the Velodyne LiDAR is very popular, but approaches using mono or stereo cameras are also prevailing. A matching algorithm is often used in tandem with a pose estimator to compute the displacement of the vehicle based on the association between consecutive observations. The main problem of the process of finding the correct correspondence between points (matching) is that it is difficult and a time-consuming task.

SLAM is an approach to localization and mapping that dates back to the work of [9], along with that of [10], among others. Despite the contributions to the core formulation of SLAM – representing the model distribution using Gaussian distributions [11] and being able to use Monte Carlo methods for the posterior estimates [12], [13], using this method to find the correct correspondences between points is difficult in implementation and computational intensity.

We propose a method that takes advantage of high-definition third party maps. They contain sparse, but accu-

Contact author

rately described (semantically and geographically) landmarks. The matching becomes a faster process, with the requirement that the fused sensor information has to be translated from a dense *occupancy grid map* to a feature map [3]. In the next sections we will describe in detail both the estimation process as well the data association mechanism.

B. Localization Context

The way the pose is modeled is application-specific, but for rigid mobile robots it usually has three degrees of freedom (two for position plus one for orientation) in the planar case and six degrees of freedom (three for position plus three for orientation) in the 3-dimensional case.

Each sensor records data points from the surrounding environment and builds a world representation based on its space coordinates and data rate. By building a coherent context, the localization function resolves these discrepancies and aligns map landmarks and sensor observations into a consistent time and space world representation. Synchronizing data from different sensors is the main task of the sensor fusion module. When sensor data is fused, the task of the localization function is facilitated and it mostly consists of performing an alignment of the fused sensor data with the local map.

There are mainly three types of localization - dead reckoning, *a priori* map localization, and Simultaneous Localization and Map Building (SLAM) - representing increasing levels of competence in pose estimation. Dead reckoning is considered to be a relative localization. Using sensors to perform dead reckoning, the calculation of the vehicle's position, at the present time, is based on its previous position, in the sensor's speed and the elapsed time.

Pose estimation with bounded uncertainty is only possible through the availability of *absolute* rather than incremental pose measurements [14]. An *a priori* map that stores the location of distinct landmarks provides such a source of absolute information. Thus, given an ability to sense its surroundings, the AV can obtain absolute pose estimates by registering sensed information with the map.

SLAM is performed by storing landmarks in a map as they are observed by the AV's sensors, using the car pose estimate to determine the landmark locations, while at the same time, using these landmarks to improve the AV pose estimate.

II. Dealing with uncertainty

In autonomous driving, the basic navigation is based on dead reckoning sensors that predict vehicle maneuvers and provide detailed information about the relative displacement of the car, independent of features in the environment. However, due to slippage and drift of the car, localization based only on relative information has an error that increases without bounds over time [8]. So, it is required to complement odometric data with absolute positioning. Due to their high frequency rate, the dead reckoning sensors are commonly used as redundant measurements to support and fuse with more complex localization techniques or sensors, which provide increasingly better quantities of information.

For absolute positioning using landmarks, laser, radar and vision based systems have been developed. The absolute information provides position measurements based on observations made from the environment. This position information is independent of previous position estimates. However, this comes at the price of higher computational costs, lower frequency and lower accuracy. Since the absolute measurements do not depend on previous position estimates, they do not suffer from unbounded error growth.

A. Sensor uncertainty

Features are obtained by fusing information from radar, Lidar and vision sensors, while the landmarks are delivered with a semantic meaning by the map provider. To improve the accuracy of the pose estimate, a relative and absolute position measurement is employed in this paper. The relative position measurements provide precise positioning information constantly, and, at certain times absolute measurements are made to correct the error in the relative measurements. Multi-sensor fusion provides techniques to combine information from different sensors for a better estimate of the car's position. This translates to the fact that while a vehicle is driving along a road, at every time step there is a prior estimate of its global pose with same uncertainty. If a vehicle detects a landmark, the estimate of the global position of this landmark is calculated by the vehicle pose estimate and the relative position of the landmark with respect to the vehicle. Therefore, the position uncertainty of a landmark is twofold: first the measurement uncertainty of the sensors and second the uncertainty of the prior vehicle pose estimate.

At any time step, t, there are likely to be multiple different measurements, $z_t = \{z_t^1, z_t^2, \ldots, z_t^K\}$. Each corresponds to an observation of a different map element, m_i, within the sensor's field-of-view of vehicle pose. The general form of the vehicle-relative observation of an object within the environment, is a nonlinear function h of the vehicle pose and the corresponding map element state, m_i:

$$z_t^j = h(x_t, m_i) + \lambda_t, \tag{1}$$

where λ_t captures the sensor noise and uncertainties. It is usually modeled as zero-mean Gaussian $\lambda_t \sim \mathcal{N}(0, \Lambda_t)$ and therefore $z_t^j \sim \mathcal{N}(h(x_t, m_i), \Lambda_t)$.

B. Ego motion uncertainty

The position and orientation of an autonomous car are critical pieces of information since they provide a link between a car's external state and the state of its environment: this allows the positions of objects and places to be expressed in the car's own coordinate frame and therefore enables the AV to navigate the environment. Since current commercial satellite based localization systems (e.g. GPS) offer neither sufficient accuracy nor availability, new methods for self-localization must be developed and applied to improve the overall localization result.

The localization function assumes that the ego vehicle model provides a prediction of the next state of the pose, which is then refined by taking into account the observations from the sensors to achieve higher accuracy. Incremental localization is a reasonably linear process and the Extended Kalman Filter (EKF) requirement of small estimation errors is generally met.

The localization process model specifies that the vehicle moves relative to its previous pose according to a dead reckoning motion estimate. Denote by $x_t \in \mathcal{R}^p$ the state of the car at time, t. The state vector describes the vehicle pose (position and orientation), and may also include linear and angular velocities. In our case the *pose* of the vehicle is comprised of its two-dimensional planar coordinates relative to an external coordinate frame, along with its angular orientation. Denoting the former as x and y, and the latter by θ, the pose of the car is described by the following vector: $x_t = \begin{pmatrix} x & y & \theta \end{pmatrix}'$.

The odometry (that is sometimes referred to as *control input*) is denoted by $U_T = \{u_0, u_1, \ldots, u_T\}$ and the relation between two successive poses and the respective odometry measurement is given by a usually nonlinear motion model f:

$$x_{t+1} = f(x_t, u_t) + w_t \tag{2}$$

In order to increase the stability and accuracy of the pose estimation, the vehicles are mostly assumed to comply with certain *motion models* which describe their dynamic behavior. The question which motion model is most suitable for describing vehicles' motions in certain scenarios has not been sufficiently answered. The most common motion model implementation assumes that the velocities and angular rates are constant from one time step to the next (constant turn rate and velocity model). This is known as the low dynamics assumption and is a reasonable assumption for vehicles traveling at low speeds over relatively even terrain.

$$x_{t+1} = f^{CTRV}(x_t, u_t) = \tag{3}$$

$$x_t + \begin{cases} \begin{pmatrix} v_t cos(\theta_t)\Delta_t \\ v_t sin(\theta_t)\Delta_t \\ 0 \end{pmatrix} & ,\omega_t = 0 \\ \begin{pmatrix} \frac{v_t}{\omega_t}(sin(\theta_t + \omega_t\Delta_t) - sin(\theta_t)) \\ \frac{v_t}{\omega_t}(cos(\theta_t) - cos(\theta_t + \omega_t\Delta_t)) \\ \omega_t\Delta_t \end{pmatrix} & ,\omega_t \neq 0 \end{cases} \tag{4}$$

The noise of the odometry sensor system is usually modeled as zero-mean Gaussian and is (approximately) captured by the additional term w_t that follows a Gaussian distribution with covariance Σ_t, $w_t \sim \mathcal{N}(0, \Sigma_t)$. Therefore, x_{t+1} is modeled as a Gaussian as well:

$$x_{t+1} \sim \mathcal{N}(f(x_t, u_t), \Sigma_t) \tag{5}$$

C. Map uncertainty

Autonomous vehicles often operate in environments whose structure is known *a priori*. This information comes in the form of a map and cause-effect relationships that describe characteristics of the environment and consist of relative and absolute measurements [13]. In general, the vehicle builds either a coarse (topological) or fine (metric) map by fusing observations of its surroundings with the help of location estimates.

In general, a probabilistic approach allows us to impose softer requirements on sensor accuracy and ego motion model. Creating probabilistic maps from a noisy laser suffers from two disadvantages: first, the intensity averages for each cell will depend heavily on which beam happens to to hit it; and second, the computed intensity variance will significantly overstate the reality [15]. Also, using poorly calibrated cameras can result in fuzzy objects and large uncertainty in the description of the feature or landmark.

How to build a probability map of an environment has been actively studied in the robotics community. The map representation must permit reliable correspondence between the information obtained from the vehicle sensors and the stored map information. First, the search for observation-to-map association needs to be efficient enough for real-time operation and, second, the association must be robust to partial views and large search-spaces. The most widespread method for building a probability map of an uncertain environment is grid-based occupancy maps [16]. Occupancy values for each grid cell are determined based on sensor reading and by applying the conditional probability of occurrence using Bayes' rule [17]. One difficulty concerning occupancy grids is data association, which can be expensive if the search-space is large.

Occupancy grids are a generic and convenient means to represent the probability of the presence of individual items in metric contexts [18]. The space around the vehicle is logically structured into discrete regions or cells that cover a small portion of the environment. Occupancy grids are well-suited for environments with dense observable structures and can be built from heterogeneous sensors such as radars, lidars, and stereo cameras.

While occupancy grids provide a dense map of the environment, so called *feature maps* are sparse and contain only the position of distinct *features* or *landmarks* in the environment. In contrast to occupancy grids, feature maps can be maintained in a single map that contains only the landmarks' coordinates and their distinctive signature.

For the purpose of this work, a two-dimensional grid map with a cell size of 0.15 m is used. The content of each cell is the occupancy probability $p \in [0, 1]$, which represents the probability of a cell being occupied by any kind of object. Details on how the grid map is created can be found in [3]. Once the grid map is available, the next step is to generate an image of occupied areas. All grid cells having an occupancy probability of $p \leq 0.5$ are reset to an identical occupancy value (e.g., 0.0).

Landmarks in the grid map are represented as a connected set of grid cells which have to fulfill certain properties. For precise localization, the position of the region should not change due to changes of the environment or varying weather condition. Another desirable property is that they have to be invariant to rotation up to certain degree. This translates to the fact that they have to represent distinct, static objects along the road which are suitable for feature retrieval (e.g., poles, trees, street lamps, etc) but should not detect unregular structures like vegetation along the road. In this contribution, an OpenCV blob detection filter is used as a landmark extractor (as seen in Fig. II.1).

D. Third party maps

High-quality map providers claim that the geometrical descriptions of the roads and lanes have an accuracy of 0.5m in terms of absolute position and 10cm at a distance of 80m in terms of relative position. The expected accuracy for a landmark's relative position at a distance of 80m is 10cm. Note that the accuracy for relative positioning depends on the method adopted to align with the reference map we measure against [19].

On the road vehicles need to be able to localize themselves, and be aware of and respond to events occurring on the road. Hence, a map requires to be updated second by second in order to serve this purpose and a constant validation and update mechanism needs to be in place, with vehicles having the ability to share and validate information about the road network.

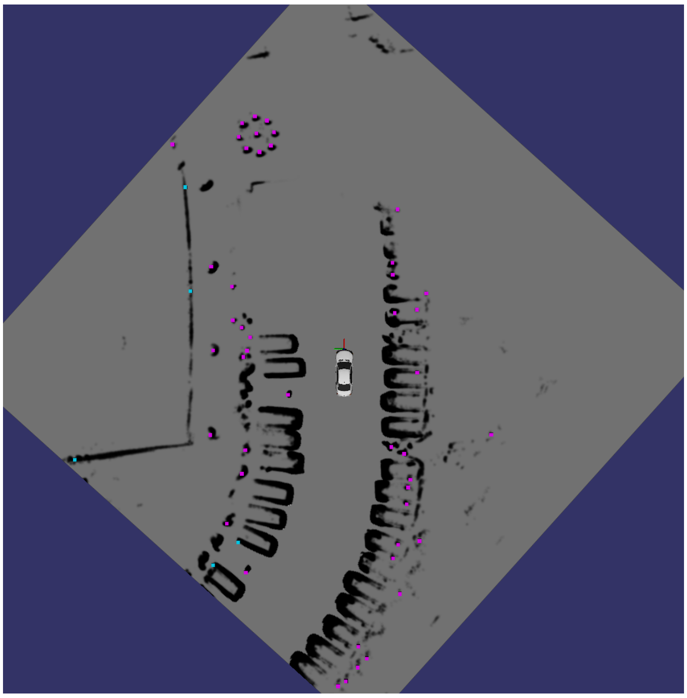


Fig. II.1: Grid map representing static objects. In magenta are the landmarks extracted using a blob detector.

III. Gated Extended Kalman Filter

The localization frameworks can be summarized as: each sensor produces a series of observations that are shared with the localization function. An observation contains a description for a detected object, including its type and position, and a timestamp. Localization and sensors have to define a common data format for observations. The localization function does not process raw data or intermediate representation generated from the sensors. It is expected that the volume of the data to be provided as input to localization is fairly small and data transfer does not present any particular technical challenges. Note that this facilitates improving the overall efficiency of the system and it enables testing localization in simulation environments where the inputs consist of sequences of pre-recorded observations. If the in-vehicle system includes a sensor fusion function, this module is responsible for passing observations for all fused sensors to localization. When observations are the end results of sensor fusion, localization expects that all observations refer to a coherent space and time representation defined by the sensor fusion module (space and time synchronization). Note that, while environment perception is an extremely complex function, for localization purposes, only persistent static information is relevant.

Since no sensor takes perfect measurements or works well in all situations, representing and operating on uncertainty with a statistical tool such as Bayes filters is key in any system using many sensors [12], [20]. Kalman-filter based techniques [8], [21] have proven to be robust and accurate for keeping track of the car's position. Because of its concise representation (the mean and covariance matrix suffice to describe the entire density) it is also a particularly efficient algorithm. One of the main drawbacks of the Kalman filter is that is does not correctly handle non-linear or non-Gaussian motion and measurement models, and can not deal with multi-modal densities. However, the non-linearities can be accommodated using non-optimal extensions of the Kalman filter (in our case EKF), while multi-modal densities are mostly encountered during global localization.

Another way to make computation tractable, is to assume that the dynamic system is a Markov process – that is, the current state variable x_t contains all relevant information. The Bayes filter shifts the belief in the direction of motion and smoothes it out to account for inherent uncertainty in motion estimates. Implementing Bayes filters requires specifying the perceptual model $p(z_t|x_t)$, the dynamics $p(x_t|x_{t-1})$, and the representation of the belief.

To localize a car we need to recursively compute the density $p(x_k|z_k)$ at each time-step. This is done in two phases:

1) *Prediction Phase* In the first phase we use a *motion model* to predict the current position of the car in the form of a predictive PDF $p(x_k|z_{k-1})$, taking only motion into account. We assume that the current state x_k is only dependent on the previous state x_{k-1} (Markov) and a known control input u_{k-1}, and that the motion model is specified as a conditional density $p(x_k|x_{k-1}, u_{k-1})$. The predictive density over x_k is then obtained by integration:

$$p(x_k|z_{k-1}) = \int p(x_k|x_{k-1}, u_{k-1})p(x_{k-1}|z_{k-1})dx_{k-1} \tag{6}$$

2) *Update Phase* In the second phase we use a *measurement model* to incorporate information from the sensors to obtain the posterior PDF $p(x_k|z_k)$. We assume that the measurement z_k is conditionally independent of earlier measurements z_{k-1} given x_k, and that the measurement model is given in terms of a likelihood $p(z_k|x_k)$. This term expresses the likelihood that the car is at location x_k given that z_k was observed. The posterior density over x_k is obtained using Bayes theorem:

$$p(x_k|z_k) = \frac{p(z_k|x_k)p(x_k|z_{k-1})}{p(z_k|z_{k-1})} \tag{7}$$

After the update phase, the process is repeated recursively. At time t_0 the knowledge about the initial state x_0 is assumed to be available in the form of a density $p(x_0)$. The Kalman gain controls whether the filter puts more trust into the prediction or the sensor measurement z_t: if K is small, then the posterior mean will be close to the predicted mean, hence the prediction is more trusted than the sensor measurements and vice versa.

IV. Data association

One of the main steps in localization is the map matching or data association. This relies on correct correspondence between data obtained from the car sensors and the data currently stored in the map. In natural environments the varied distribution of landmarks requires a robust data association algorithm that can accommodate both high landmark density and an absence of stable landmarks. In our case, a map of the environment is a list of objects in the environment and their locations.

Data association is arguably the main weakness of feature map localization. However, data association failure is a much more serious problem for SLAM than for *a priori* map localization. Simple localization may be able to recover from a minor mis-association because only the vehicle pose estimate is affected, but with SLAM the map is also altered and these inconsistencies tend to be self-propagating, causing divergence from the true pose.

The most popular method for solving data association problem is the Nearest Neighbor (NN). This method aim is establishing the most likely pairing by minimizing the squared Mahalanobis distance between the observations and their associated predictions. We use a NN for outlier detector and a *maximum likelihood estimator (MLE)* to find the best mach between the observation and the map when there is a high level of ambiguity.

A. Gating validation

Since, within the EKF framework, the position of both the object in the map and the observed landmark are assumed to have Gaussian uncertainty, it is possible that any given observation might correspond to any target. However, to reject unlikely associations, only landmarks within a reasonable neighborhood of an observation should be considered. The Mahalanobis distance is defined as: $M_{ij} = \mu_{ij}^T S_{ij}^{-1} \mu_{ij}$, where $\mu = z_i - h(\hat{x}_j)$ is the observation innovation with covariance S_{ij}. For a Gaussian distributed innovation sequence, the Mahalanobis distance, also known as the *normalized innovation squared (NIS)* forms a χ^2 (chi-squared) distribution. The gate is applied as a maximum NIS threshold $M_{ij} < \gamma$.

The threshold γ allows us to quantify how likely measurement of a target are to fall within the validation gate. A large process noise covariance can partly compensate for a poor motion model, however, large process noise causes the validation gate to be large which in turn increases the level of ambiguity for data association.

In this paper, we are using a validation gate in combination with the most common ambiguity resolution method – *nearest neighbor* data association. Given a set of observations $z_t = \{z_t^1, z_t^2, \ldots, z_t^K\}$, within the validation gate of target x, a likelihood of association L_i can be calculated for each $z_t^i \in z_t$:

$$L_i = \frac{1}{det(2\pi S_t^k)^{\frac{1}{2}}} \exp\{-\frac{1}{2}(z_t^i - \hat{z}_t^k)^T [S_t^k]^{-1} (z_t^i - \hat{z}_t^k)\} \quad (8)$$

Nearest neighbor data association then selects the observation that maximizes L_i. This is sometimes referred to as *maximum likelihood* data association.

V. Results

Today, GPS, IMU, odometer, and encoders are used extensively to estimate the vehicle's state, while Lidar, radar and cameras are integrated to generate reliable information of terrain, obstacles, pallets, pedestrians, and the surrounding environment. The accuracy and availability of GPS prevent it from being the sole means of measuring location for autonomous navigation of automobiles. In their 2014 GPS Performance Analysis Report, the FAA reported horizontal accuracies ranging from 2.192 to 8.465 meters with 95% confidence and horizontal accuracies with 99.99% confidence ranging from 4.595 to 16.851 meters [22]. However, for an autonomous vehicle to stay in a lane, the localization requirements are in the order of decimeters. Thus, the localization cannot reliably be achieved using GPS-based inertial guidance systems, particularly in urban areas [15].

The techniques for fusing GPS and IMU data are typically categorized as *tightly* or *loosely* coupled [23]. A loosely coupled system uses GPS as a black box which generates positional and velocity information. This information is then fused with IMU acceleration and integrated velocity and position terms to generate an overall state estimate. A tightly coupled system, in contrast, uses the pseudorange measurements as direct inputs into the filter, and solves for vehicle state and dynamics estimate in an integrated manner. Our vehicle is equipped with a tightly coupled system, having tested units provided by navigation distributors such as iMAR and NovAtel. However, this is not a complete solution for most urban environments, wherein GPS availability is typically discontinuous. As a result, the output of an inertial system is used as a reference, but it is not equivalent with a "ground truth" pose information.

Figure V.1 shows the offline recordings of iMAR position trace and the constant turn rate and velocity (CTRV) model output within a parking lot, chosen for our convenience. After more than one drive around the parking lot, the drift in the model increased, resulting in a $\approx 2m$ difference at the end of the drive. This drift can be corrected using sensor observations in a filtering manner.

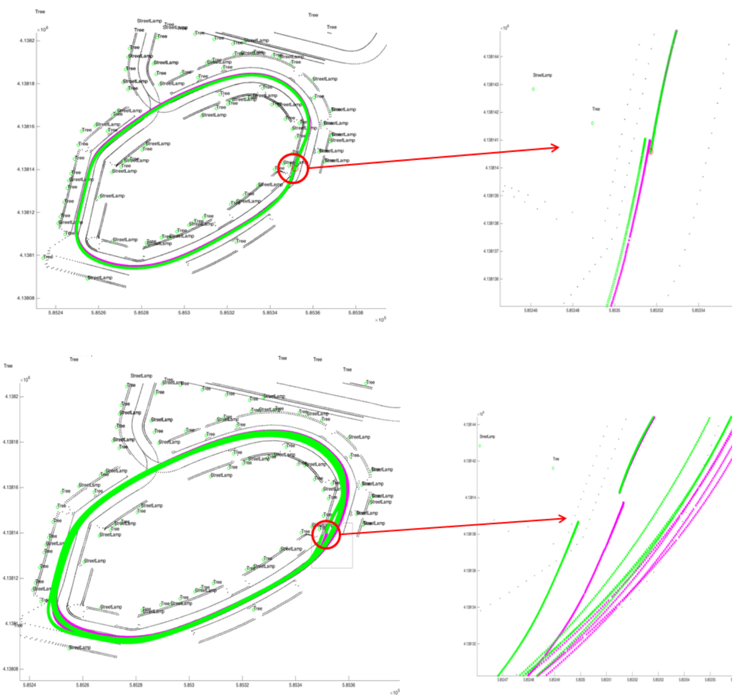
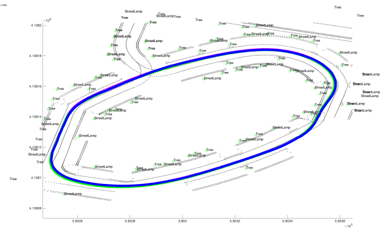
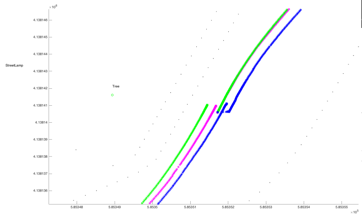


Fig. V.1: iMAR trace in magenta, CTRV green, offline map landmarks in blue squares, dotted lines represent the road information.The starting location is marked as the red circle.

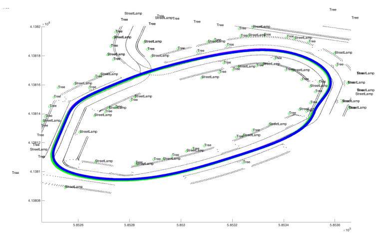
With the introduction of the Extended Kalman Filter to account for the uncertainties in both the process and sensors measurements, along with a robust data association between the offline (provider) map and online observation, we are able to correct the odometry results and keep the vehicle on the road. Since the third party provider map contains discrete landmarks with semantic information, one of our challenges was to transform a dense grid map into a layer that could be overlaid on top of the provider map. This was achieved by using a blob detector and the center of each blob was label as the center of an online feature. A standard EKF has been employed and compared with a gated NN outlier detector EKF, as seen in Figure V.2 and Figure V.3. While, for a single drive around the parking lot, the distance between the two EKF methods and iMAR trace at the ending location is within the same order of magnitude ($22.3cm$ for standard EKF and $24.8cm$ for gated EKF), the error is significantly higher when we drive multiple times. In this case, the standard EKF performs poorly, and with no correct association between landmarks and observations the distance to the iMAR is $18.2m$. However, the gated EKF is constantly correcting the drift in the odometry and the correction are performed such that the influence of outliers is either ignored or has a very small weight in the result of the final pose. Figure V.4 shows the online results of the localization algorithm, where "almost" perfect association between the map's landmarks and online observation are in magenta circles.



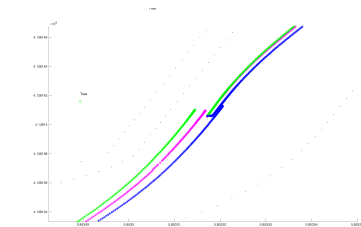
(a)



(b)

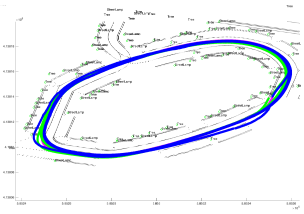


(c)

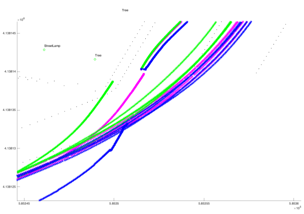


(d)

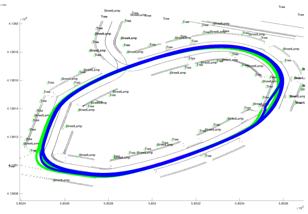
Fig. V.2: a) One drive Standard EKF b) One drive standard EKF - zoom at the staring and ending location c) One drive gated EKF d) One drive gated EKF - zoom at the staring and ending location.



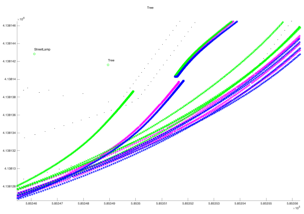
(a)



(b)



(c)



(d)

Fig. V.3: Three drives around the parking lot: a) standard EKF b) standard EKF - zoom at the staring and ending location c) gated EKF d) gated EKF - zoom at the staring and ending location.

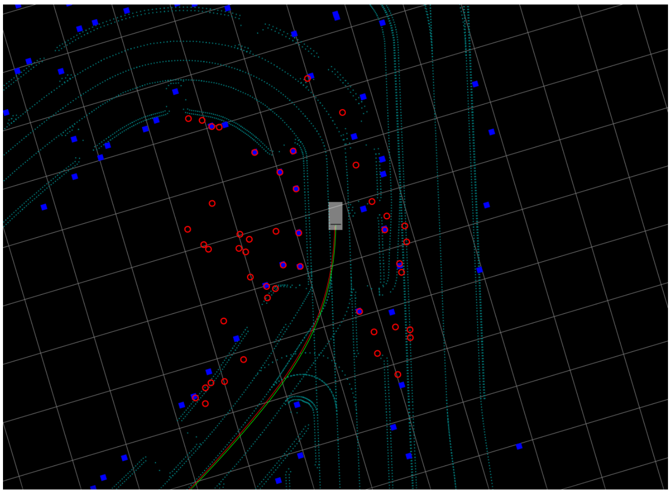


Fig. V.4: Online localization results: GPS trace in red, localization pose estimate in green, offline map landmarks in blue squares, in red circle are the online features, dotted lines represent the road information.

VI. Conclusions

Localization is a critical enabling component of autonomous vehicle navigation. In this paper we present our approach to localization for autonomous cars by using an improved EKF algorithm on a third party feature map. The change in the uncertainty of a map feature is dependent on the car's path and the uncertainty of its relative motion estimates. Our approach has been implemented and tested using a real car equipped with different types of sensors. All experiments presented in this paper show the result of real world data obtained with this car. Experiments based on measured data have shown that the recognition of features points solely by position strongly depends on the prior vehicle position uncertainty.

Acknowledgements

The authors would like to thank Christoph Rothermel for insightful discussions and feedback.

References

[1] I. Skog and P. Händel, "In-car positioning and navigation technologiesa survey," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 10, no. 1, pp. 4–21, 2009.

[2] J. Levinson, M. Montemerlo, and S. Thrun, "Map-based precision vehicle localization in urban environments." in *Robotics: Science and Systems*, vol. 4. Citeseer, 2007, p. 1.

[3] J. Wiest, H. Deusch, D. Nuss, S. Reuter, M. Fritzsche, and K. Dietmayer, "Localization based on region descriptors in grid maps," in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*. IEEE, 2014, pp. 793–799.

[4] M. Rapp, T. Giese, M. Hahn, J. Dickmann, and K. Dietmeyer, "A feature-based approach for group-wise grid map registration," in *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*. IEEE, 2015, pp. 511–516.

[5] S. Thrun, W. Burgard, and D. Fox, "A probabilistic approach to concurrent mapping and localization for mobile robots," *Autonomous Robots*, vol. 5, no. 3-4, pp. 253–271, 1998.

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

[7] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part i," *Robotics & Automation Magazine, IEEE*, vol. 13, no. 2, pp. 99–110, 2006.

[8] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte carlo localization for mobile robots," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 2. IEEE, 1999, pp. 1322–1328.

[9] G. Giralt, R. Sobek, and R. Chatila, "A multi-level planning and navigation system for a mobile robot: a first approach to hilare," in *Proceedings of the 6th international joint conference on Artificial intelligence-Volume 1*. Morgan Kaufmann Publishers Inc., 1979, pp. 335–337.

[10] P. Cheeseman, R. Smith, and M. Self, "A stochastic map for uncertain spatial relationships," in *4th International Symposium on Robotic Research*, 1987, pp. 467–474.

[11] M. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (slam) problem," *Robotics and Automation, IEEE Transactions on*, vol. 17, no. 3, pp. 229–241, 2001.

[12] D. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello, "Bayesian filtering for location estimation," *IEEE pervasive computing*, no. 3, pp. 24–33, 2003.

[13] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust monte carlo localization for mobile robots," *Artificial intelligence*, vol. 128, no. 1, pp. 99–141, 2001.

[14] T. Bailey, "Mobile robot localisation and mapping in extensive outdoor environments," Ph.D. dissertation, Citeseer, 2002.

[15] 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.

[16] S. P. Engelson and D. V. McDermott, "Error correction in mobile robot map learning," in *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*. IEEE, 1992, pp. 2555–2560.

[17] 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.

[18] A. Elfes, "Occupancy grids: A stochastic spatial representation for active robot perception," *arXiv preprint arXiv:1304.1098*, 2013.

[19] L. Kent. (2015) Autonomous cars can only understand the real world through a map. [Online]. Available: http://360.here.com/2015/04/16/autonomous-cars-can-understand-real-world-map/

[20] K. Werber, M. Barjenbruch, J. Klappstein, J. Dickmann, and C. Waldschmidt, "Roughcougha new image registration method for radar based vehicle self-localization," in *Information Fusion (Fusion), 2015 18th International Conference on*. IEEE, 2015, pp. 1533–1541.

[21] L. Moreno, J. M. Armingol, S. Garrido, A. De La Escalera, and M. A. Salichs, "A genetic algorithm for mobile robot localization using ultrasonic sensors," *Journal of Intelligent and Robotic Systems*, vol. 34, no. 2, pp. 135–154, 2002.

[22] W. J. Hughes, "Global positioning system (gps) standard positioning service (sps) performance analysis report," *Tech. Cntr. NSTB/WAAS T and E Team*, no. 87, 2014.

[23] S. Kennedy, D. Cosandier, and J. Hamilton, "GPS/INS integration in real-time and post-processing with novatel's span system," in *Proceedings of the International Global Navigation Satellite Systems Society Symposium 2007*, 2007, pp. 4–6.