sensors

MDPI

Article

Improved LiDAR Probabilistic Localization for Autonomous Vehicles Using GNSS

Miguel Ángel de Miguel *, Fernando García and José María Armingol

Intelligent Systems Laboratory, Universidad Carlos III de Madrid, Av. de la Universidad, 30, 28911 Leganés, Madrid, Spain; fegarcia@ing.uc3m.es (F.G.); armingol@ing.uc3m.es (J.M.A.)

* Correspondence: mimiguel@ing.uc3m.es

Received: 27 April 2020; Accepted: 30 May 2020; Published: 2 June 2020

Abstract: This paper proposes a method that improves autonomous vehicles localization using a modification of probabilistic laser localization like Monte Carlo Localization (MCL) algorithm, enhancing the weights of the particles by adding Kalman filtered Global Navigation Satellite System (GNSS) information. GNSS data are used to improve localization accuracy in places with fewer map features and to prevent the kidnapped robot problems. Besides, laser information improves accuracy in places where the map has more features and GNSS higher covariance, allowing the approach to be used in specifically difficult scenarios for GNSS such as urban canyons. The algorithm is tested using KITTI odometry dataset proving that it improves localization compared with classic GNSS + Inertial Navigation System (INS) fusion and Adaptive Monte Carlo Localization (AMCL), it is also tested in the autonomous vehicle platform of the Intelligent Systems Lab (LSI), of the University Carlos III de of Madrid, providing qualitative results.

Keywords: localization; LiDAR; GNSS; Global Positioning System (GPS); monte carlo; particle filter; autonomous driving

1. Introduction

Autonomous vehicle localization is the problem of estimating its position, determined by the x and y coordinates in a map and its orientation. This localization must be as accurate as possible since many vehicle's modules, such as control or path planning strongly depend on how good it is. Errors in localization can cause the vehicle to have an undesirable behavior or to even not being able to follow the desired path. Localization techniques can be divided mainly in mapping or sensor based [1]. Global Navigation Satellite System (GNSS) information is commonly used to solve localization problems using a sensor. It provides a good global localization with no drift but it has to deal with some errors from different sources. Those errors can be generated due to the satellites themselves (e.g., clock inaccuracies or dilution of precision), interference in the satellite signal (e.g., signal jamming, satellite occlusion) or signal propagation errors. That last error source includes inaccuracies produced by different weather conditions in the ionosphere and troposphere earth layers, and by multipath interference, caused by the reflection of the satellite waves when the vehicle is surrounded by large obstacles (high buildings or trees) [2,3].

High precision GNSS receivers, like Real Time Kinematic (RTK) or differential GPS, improves accuracy considerably but they have commonly a very high cost and they don't solve some accuracy errors. On the other hand, laser probabilistic localization methods, like Monte Carlo Localization (MCL) [4] can compare a precomputed map with the laser readings to acquire a precise position and orientation of the vehicle. The problem presented by this kind of algorithms is the opposite that with GNSS; in open environments with less map features, their accuracy decreases significantly. Here, particles of the filter would disperse generating different clusters of particles far away from the actual localization. This is known as the kidnapped robot problem and is a common localization error for probabilistic methods like MCL [5].

To solve those autonomous vehicle's localization problems, many different solutions have been proposed. One of the most known solution includes the fusion of different localization sources using Kalman filter based methods, i.e., Extended Kalman Filter or Unscented Kalman Filter [6]. This method commonly fuses GNSS, IMU and other different odometry sources like the one generated by the wheels encoders or LiDAR/camera odometry.

Fusion filters based on Kalman fuses all the localization sources and generates a more accurate estimation of the position, but they depend strongly on the accuracy of the GNSS measurements, as it is the only absolute localization source i.e. the only one that does not have drift.

Furthermore, GNSS can suffer different problems, being one of the most common the multipath problem, which is a common reaserch topic where several works try to reduce it. In [7] , a digital map of the environment is generated with OpenStreetMaps to prevent these errors and in [8], multipath is estimated and mitigated using a particle filter. However, the results don't give enough accuracy and they depend on the information in OpenStreetMaps which is not always available.

Another solution adopted by some researchers consists in probabilistic localization, which can be performed using different sensors such as LiDAR [9], cameras [10,11] or magnetometers [12]. One of the most common method for this kind of solution is Monte Carlo Localization (MCL) [4]. MCL works as a probabilistic particle filter that uses the match between the LiDAR sensor and the map as a feature for each particle that determines the probability of existing in the next iteration. As an improvement for MCL, Adaptive Monte Carlo Localization (AMCL) outperforms classic (MCL) [13] as it uses Kullback-Leibler Distance (KLD) sampling to make the filter converge faster. Particle filter localization methods can be applied to autonomous vehicles, like in [9], that uses a 2D LiDAR and a map with the features of the road.

Both previously commented algorithms, GNSS and MCL have a good performance, but also have some drawbacks. To get the best from every algorithm, several works explore the idea of combining the two sources of information to improve the localization. Most of the works available in the literature, improve MCL by resetting the calculated position when it differs too much from the GNSS position, in other words, the kidnapped robot problem is corrected once it is detected. Those improvements consists of resetting methods, which makes MCL more robust. An example of this is [14], where the kidnapped robot is detected and then solved using the Expansion resetting method.

Different localization sensors are used to solve this problem like in [15], where GNSS is used to detect and solve it, or in [16], where WiFi signal detection provides information about the localization error. These methods give good results as they are able to detect and solve the kidnapped robot error in most of the cases. However, the time necessary to detect and solve this problem adds further error to the system, as during this time, the vehicle is driving with a wrong localization. That localization error is unacceptable for autonomous driving vehicles as it would result in wrong control or path planning commands and thus, incorrect behavior. Consequently, it seems a reasonable assumption that the prevention of the kidnapped problem, as it is intended in this work, would lead to better results.

The work of [17] gives a solution based on replacing particles with a low degree of laser fit on the map with new particles according to the probability density given by the sensor, but it can have localization problems in empty maps as no GNSS is used. In [18], GNSS data are used to generate new particles on the filter and to eliminate distant ones. However, this method doesn't handle orientation, it only considers the position. Furthermore introducing particles based on GNSS data in all the cycles of the filter, would increase the noise of the localization and unfortunately no quantitative results of its performance are provided. As shown in [19], fusion of both sensors based on particle weight gives better results than adding new particles based on GNSS data. However, that work does not handle the kidnapped robot problem, as no strategies to recover are performed when the GNSS and particle filter positions differ considerably.

In contrast to all the approaches mentioned above, our method continuously uses GNSS data in the filter in both ways: modifying the weight of the particles, and injecting new ones if needed, avoiding kidnapped robot problem and making unnecessary to detect and reset states where the robot is badly localized. Furthermore, the method is designed to not replace directly the particles with new ones based on GNSS data, but to calculate its probability considering multiple parameters of both localization sources, making the particle filter more stable, an reducing noise generated by adding directly new particles on every cycle of the filter. Besides, we also consider orientation error when calculating the new probability, which is not done in most of the reviewed works.

The rest of the paper is structured as follows: Sections 2 and 3 describe the software architecture and the method proposed respectively, Section 4 evaluates the localization with real data and in Section 5 the conclusions are exposed.

The proposed method is coded and tested with the KITTI Dataset [20]. The source code of this method is publicly available at https://github.com/midemig/gps_amcl so anyone can replicate the experiments described in this work.

2. Sensors and Software Architecture

This section describes the software architecture of the tests, including the configuration of the different modules used. This architecture includes the AMCL implementation for Robot Operating System (ROS) [21] from [4], the GNSS/INS based localization module [22] and the map generation module, in charge of the generation of the map, used by AMCL module.

2.1. Software Modules

In this section, the specific version of the software modules used in the comparison are described.

2.1.1. AMCL

This algorithm outperforms original MCL [13] and is chosen to be the probabilistic LiDAR localization used. Specifically, it is used the AMCL ROS node implementation as it is a stable and maintained version of the algorithm. Most of the parameters' configuration is set to the default values values, however, the most relevant ones are shown in Table 1, were the odometry model defines the equations that better describe the movement of the vehicle, the laser model describes the method used to calculate the probability of being at a certain position given a laser measurement and max particles and beams represents the maximum number of possible positions and laser beams around the vehicle respectively (more particles and beams can increase localization accuracy, but also computation time).

Table 1. Adaptive Monte Carlo Localization (AMCL)'s most relevant parameters.

Parameter	Value
Odometry model	differential
Laser model	likelihood filed
Max particles	2000
Max beams	360

When the AMCL module is used in a map with a small number of reference obstacles surrounding the vehicle, the kidnapped robot localization error can easily appear, as shown in Figure 1, where multiple particles clusters appear due to the lack of features in the map, and the localization obtained by AMCL is inaccurate.

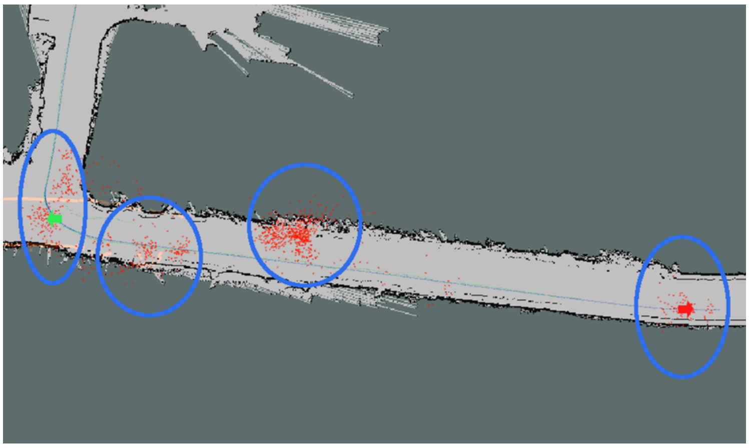

Figure 1. AMCL kidnapped robot localization error. Different clusters rounded in blue, red dots are the filter's particles and red and green arrows are ground truth.

2.1.2. Fusing Method

Kalman filter is commonly used for fusing localization data from different sources giving, as a result, a more accurate one. The software version used in this work is the robot localization ROS implementation [22]. Unscent Kalman filter is used as it is known to deal better with non-linearities in the filtering process [23]. In order to improve GNSS localization, it is fused with IMU data.

Robot localization package can fuse n different localization sources enhancing single-sensor based localization, and providing useful information such as the covariance of the position, which is of great relevance in this work.

2.1.3. Map Generation

To use AMCL localization, a pre-generated map of the environment is generated using different Simultaneous Localization and Mapping algorithms [24]. As later on, GNSS information will be added to the localization step, this map must be generated with GNSS localization data. That means that a high precision GNSS receiver is needed, but only for the map generation. The Laser information is transformed accordingly to the GNSS position at each time step and then accumulated into the map using gmapping Simultaneous Localization and Mapping (SLAM) method [25]. The generated map (Figure 2) has all the features needed later by the AMCL algorithm to match a new laser scan with it.

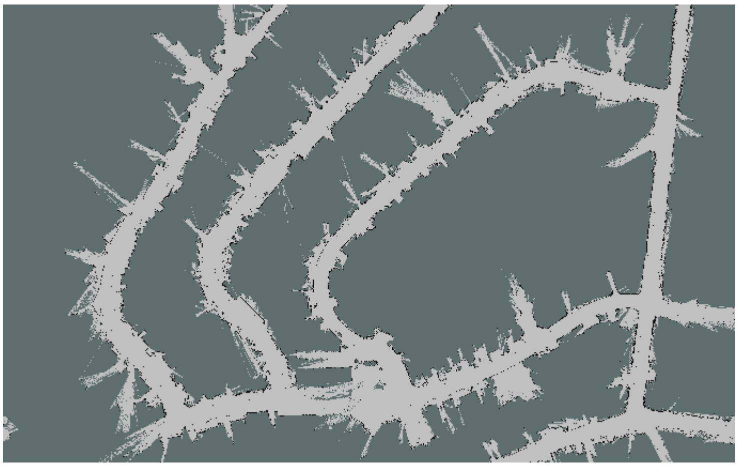

Figure 2. Map generated of one of the KITTI sequences.

2.2. Vehicle Sensor Setup

Although the results presented in the test section were performed using KITTI dataset, this algorithm can be implemented in any vehicle that fulfills some specified sensor requirements that are explained in this subsection:

2.2.1. LiDAR

LiDAR Information is necessary to provide map information, although other 3D output may be used, the recommendation for this application, where accuracy is a key point, it is to use 3D Laser scanner technology. KITTI data provides a 64 layers 3D laser scanner. As 360 degrees 2D LiDAR scanner is needed to get environment information and match it with the pre-generated map using AMCL, the 3D laser scanner information is converted to 2D. This is done by choosing a minimum and a maximum height from the LiDAR, the multiple layers can be converted into a single 2D one. These parameters are selected to incorporate the maximum amount of features from the environment, but removing the ground plane, as it would only increase the noise in the particle filter. Other LiDAR configuration can be used, the test vehicle of LSI where this algorithm was implemented and tested is based on a 32 layers LiDAR, which provide similar results.

2.2.2. GNSS Receiver

For this application, a Low-cost GNSS receiver can be used. Instead of high-cost RTK or differential GPS receivers, our method can work with lower accuracy in GNSS localization, making possible the generalization of these applications [26]. As the only localization information provided by KITTI dataset is the ground truth, it is considered to not have any zero error. Here, a low-accuracy GNSS receiver can be simulated by adding Gaussian noise to every ground truth measure, providing output similar to low-cost sensors [27]. The qualitative results provided on this test were performed using a low-accuracy GPS receiver, based on PixHawk technology.

2.2.3. Inertial Measurement Unit (IMU) Sensor

IMU is the most common sensor fused with GNSS data, as it can provide position and orientation data. KITTI database provides IMU data with extrinsic calibration information, needed for the fusion. The PixHawk sensor unit used in the qualitative tests were performed using the PixHawk unit.

3. Method Description

Based on the original AMCL algorithm, several modifications are made to integrate GNSS data into the loop.

In this section, all those modifications are detailed and justified. Furthermore, the resulting algorithm is presented in Algorithm 1.

3.1. LiDAR Likelihood

The proposed solution computes a weight for each particle in AMCL by comparing the laser data transformed to each particle position with the map. In addition to that weight, our method computes a score of how accurate this particle is matched with the map. This score s_i is computed using a Gaussian model [28,29] for the LiDAR data, following the expression:

$$s_i = \frac{1}{N} \cdot \sum_{n=1}^{N} \frac{1}{\sigma_{hit} \cdot \sqrt{2 \cdot \pi}} \cdot e^{\frac{-z_n^2}{2 * \sigma_{hit}^2}} \tag{1}$$

where N is the number of lasers of a laser scan, z is the distance from the laser hit point to the closest map occupied cell and σ_{hit} is the standard deviation of the laser. With this expression, we can evaluate how good the LiDAR measurements match with the map in every particle position, and is later used to modify the probability of that particle to exist.

3.2. GNSS likelihood Estimation

In addition to weight calculated based on the matching of the sensor with the map, we add a second weight based on the GNSS Kalman filtered position estimation. As for the LiDAR likelihood, a Gaussian model is used to estimate the GNSS based weight of each particle d_i, but in this case, as the position received and each particle of the filter are three dimensional (x, y and ψ), the n dimensional Gaussian model is used,

$$d_i = \frac{1}{(2 \cdot \pi)^{\frac{3}{2}} \cdot |\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k)\right) \tag{2}$$

where the received position is $\mu_k = (x_k, y_k, \psi_k)$ with covariance matrix Σ_k and the position and orientation of each particle is defined with $x_i = (x_i, y_i, \psi_i)$. Using this model, orientation error is also considered when computing the GNSS based weight. Furthermore multiplying all the errors by the inverse of the covariance matrix in the exponential part, makes position and orientation errors scale-invariant. Here it is important to remark that the use of the covariance of the Kalman output allows reducing the importance of this weight when the quality of the information is low.

3.3. New Particle Weight Computed

To modify the weight of every particle so that Kalman filtered GNSS data are incorporated into the filter, the new weight of every particle is calculated by making use of the weights computed in (1), (2), and the following equation:

$$w_{i-new} = w_i \cdot s_i \cdot k_l + d_i \tag{3}$$

where w_i, the weight calculated by original AMCL, is modified in order to incorporate GNSS probabilistic data. The weight k_l is a constant and it is added to balance the importance of each source of information. It is empirically set to 200, where it gives the best results in all the environments tested. After this weight modification, they are normalized to make the sum of all the weights equal 1.

3.4. New Particles Generation

As the particle filter eliminates particles accordingly to its probability, the original AMCL method adds new particles randomly distributed in the map, based on two parameters that define how often it is needed to add those particles. A different function was defined to generate new particles X_{new} near GNSS Kalman position and to determine the probability of generating those new particles. The following expression defines how the new particles are generated to follow a normal distribution centered in μ_k with covariance Σ_k.

$$\begin{pmatrix} x_{new} & y_{new} & \psi_{new} \end{pmatrix}^T = \lambda_k^{\frac{1}{2}} \phi_k \cdot R + \mu_k \tag{4}$$

where R is a random vector with distribution $N[0,1]$ and λ, ϕ are the diagonal matrix of eigenvalues and the eigenvectors matrix of the covariance Σ_k respectively. Besides, the probability p of generating a new GNSS based particle at each cycle is determine as:

$$p(X_{new}|x,\mu) = \begin{cases} d_{mean}, & \text{if } d_{mean} > 0 \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

with

$$d_{mean} = p_{max} - \frac{1}{N} \cdot \sum_{n=1}^{N} d_i \tag{6}$$

where p_{max} is the maximum probability allowed to generate new particles at each filter cycle and is experimentally set to 0.01.

The addition of particles does not increase the noise in the final localization given by the particle filter, as they are only added when the filter particles begin to considerably differ from the GNSS localization, and the newly generated number never goes beyond the limit of 1%.This situation avoids the creation of GNSS based particles when the filter provides accurate detection, these are generally particularly difficult situations for the GNSS such as urban canyons.

Algorithm 1: New weight and resample of filter particles

```
Input: $x, \mu_k, w, z$
Output: $x_{new}$
for $i = 0$ to $N$ do
    $s_i = lidarLikelihood(z_i);$
    $d_i = gnssLikelihood(x_i, \mu_{ki});$
    $w_{i-new} = w_i \cdot s_i \cdot k_l + d_i;$
end
$w_{new} = normalizeWheigt(w_{new});$
for $i = 0$ to $N$ do
    $p(X_{new}|x, \mu) = probOfNewParticle(d_i);$
    if $rand() < p(X_{new}|x, \mu)$ then
        $x_{i-new} = generateNewParticlePosition(\mu_k);$
    else
        $x_{i-new} = sampleParticle(x, w_{new})$ ;          // Original function from AMCL
    end
end
```

4. Experimental Results and Discussion

Two different group of experiments were performed. On the one hand, the proposed method is tested using the KITTI odometry dataset for quantitative results. On the other hand, the LSI platform for autonomous driving is used for qualitative results [30].

4.1. Dataset

The KITTI dataset is commonly used to test different odometry algorithms, and compare them as it includes calibrated data from different sensors (cameras, LiDAR and IMU) and precise ground truth for localization. The tests performed using this dataset compare the ground truth localization accuracy with the three following methods:

- Kalman filtered GNSS and IMU. As covariance of GNSS localization is set fixed, the mean error value is displayed for visualization purpose.
- AMCL original implementation.
- Proposed method, having the same odometry source as original AMCL and the same parameters configuration.

For every position and orientation, euclidean distance and orientation error are compared with the closest one in time of the ground truth (interpolating if necessary). KITTI dataset gives multiple sequences in different scenarios. For this evaluation residential sequences were selected, since they combine narrow and empty streets, including both types of scenarios. For every sequence, first, a map is generated using the localization ground truth and the LiDAR data described in Section 2.1.3. The experiments are designed to compare the proposed method with the other two considering the worst-case scenario for our method where, at least, it needs to give similar results to the comparing methods. Then a normal situation is tested to quantify the improvements of the proposed method.

Considering this, the following three scenarios are tested.

4.2. Empty Map

This scenario refers to situations with a low number of obstacles, i.e. lack of references for the LiDAR points to match. This is considered to be the worst scenario for AMCL. However, GNSS has better accuracy when it is not surrounded by obstacles that interfere with observations. Localization error is compared for Kalman filtered GNSS and the proposed method. As it is shown in Figure 3, localization error provided by our approach is very similar to the GNSS based, giving almost the same values for position and orientation. This is according to the expected as the proposed method can not improve localization accuracy with no laser information, but it proves that in empty environments it would perform as good as GNSS based localization.

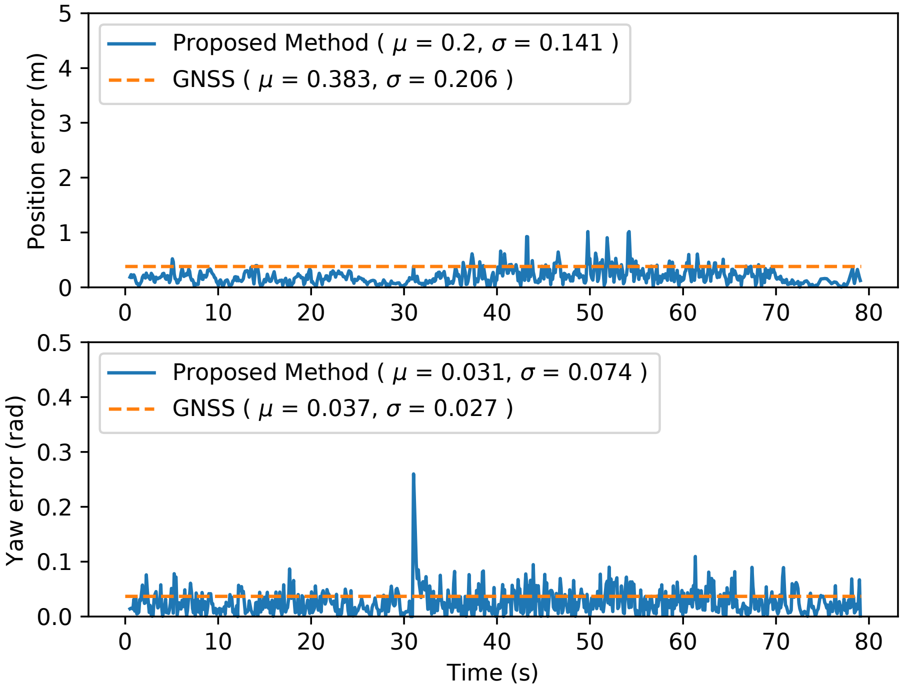

Figure 3. Comparison between proposed method with no LiDAR information but Global Navigation Satellite System (GNSS) localization. (GNSS is plotted as constant of mean value for visualitation purpose.)

4.3. GNSS Challenging Scenarios

This scenario describes the opposite problem. In an environment full of objects or an urban environment with urban canyons, GNSS localization would fail or would give noisy measurements with high covariance. In these cases, AMCL algorithm gives better results thanks to a map with numerous features and objects to match with the laser points. Localization is compared in this scenario for AMCL and the proposed method. As shown in Table 2, localization errors are similar. When no GNSS localization is received, or it has a high covariance, the proposed method, as it is designed, has a very similar behavior as original AMCL.

Table 2. Comparison between proposed method and AMCL with no GNSS information.

Method	Position Mean (m)	Position Std (m)	Yaw Mean	Yaw Std
Proposed	0.513	0.856	0.033	0.025
AMCL	0.566	0.742	0.023	0.023

4.4. Mixed Environments

The last scenario tested is a more common one where GNSS data are received with an acceptable covariance, and the map has features unequally distributed on it, generating places with more objects and empty places. The three algorithms are tested and compared in this scenario using the KITTI dataset residential sequences, that include more than 45 min of recorded data in a residential environment, but only the most relevant ones are discussed in this section.

4.4.1. AMCL

When evaluating AMCL performance two different localization issues can be identified. The first one, as shown in Figure 4, presents an increase in the error due to multiple particles' clusters that make the filter jump from one to another, increasing the error but finally converging again to the true localization. The second problem can be seen in Figure 5, where the kidnapped robot problem appears around the second 180, where the localization jumps to a similar place in the map.

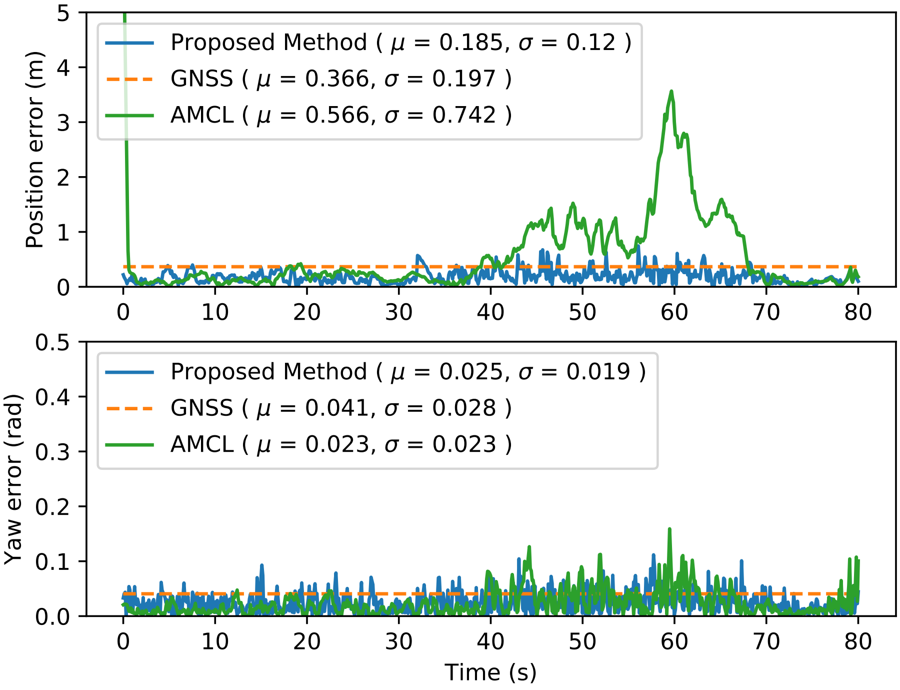

Figure 4. Localization error of the three methods described in KITTI sequence 36. (GNSS is plotted as constant of mean value for visualitation purpose).

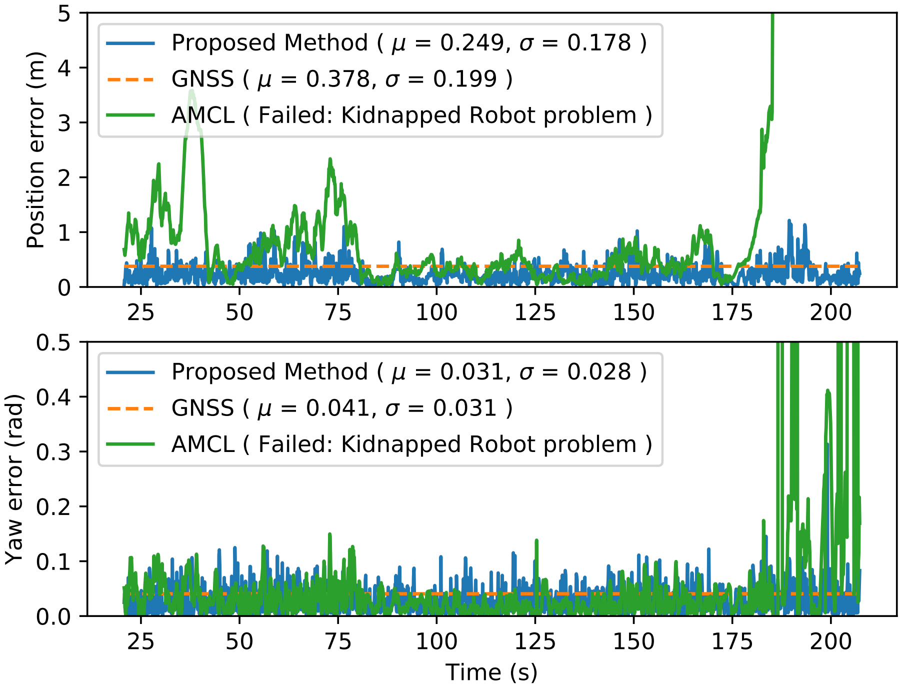

Figure 5. Localization error of the three methods described in KITTI sequence 34. (GNSS is plotted as constant of mean value for visualitation purpose).

4.4.2. GNSS

It is set to have a constant covariance error, so this localization maintains the same localization error along all the sequence. In real environments, Kalman results could be even worse as GNSS position error could be even higher in narrow streets.

4.4.3. Proposed Method

The proposed method, gives a better performance along all the path, improving localization mean error compared to AMCL and GNSS. It outperforms the other algorithms in terms of accuracy and stability as it has a localization error better or equal to Kalman localization method, and does not present kidnapped robot localization problem.

Figures 4 and 5 present the localization error along the sequence and the mean error values for sequences 36 and 34 respectively, showing how the proposed method outperforms original AMCL avoiding the kidnapped robot problem, while maintaining the localization error below the GNSS one.

Moreover, Table 3 compares the error of the proposed method as the accuracy of the GNSS localization decreases. As it is shown, for high GNSS accuracy, the error is close, but as the GNSS error increases, our method is able to reduce it using the laser information, giving always better results than the original AMCL algorithm.

Table 3. Evaluation of the proposed method errors for different GNSS mean errors in sequence 34.

GNSS Mean (m)	Position Mean (m)	Position Std (m)	Yaw Mean	Yaw Std
0.127	0.141	0.137	0.015	0.016
0.374	0.186	0.11	0.028	0.024
1.256	0.367	1.767	0.029	0.129
6.256	0.496	1.963	0.023	0.116
12.600	0.554	2.115	0.026	0.151
37.581	0.593	2.495	0.032	0.196
AMCL		Robot kidnapped error		

4.5. Real Platform Qualitative Results

In addition to the evaluation using KITTI database, the proposed method is also tested in our research platform as shown in Figure 6. As this platform does not include ground truth localization information, only a qualitative analysis of the performance of the proposed method is possible giving, as a result, a stable localization accurate enough to control the vehicle, solving the kidnapped robot that this platform showed before the implementation of this method when using AMCL, and improving the localization accuracy of the GNSS receiver.

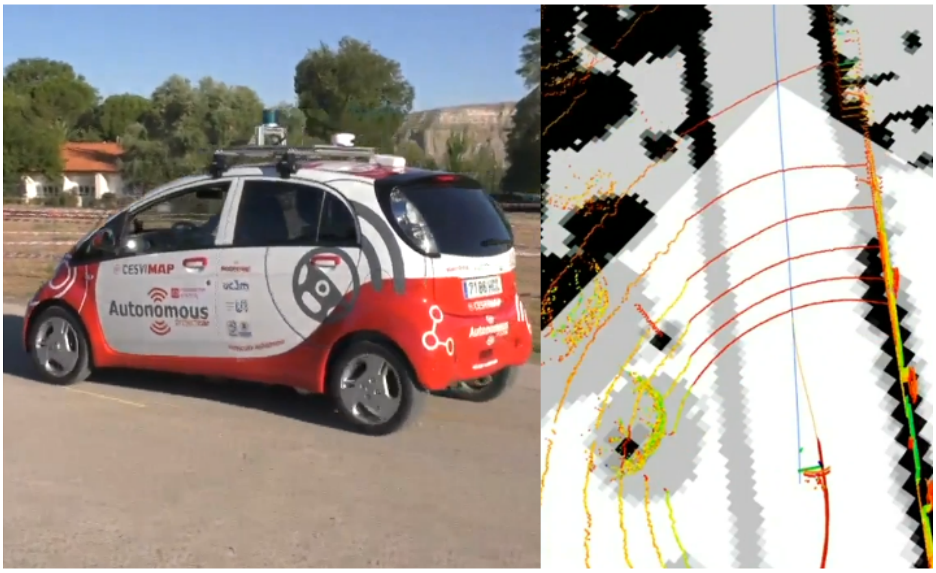

Figure 6. Our autonomous vehicle platform testing the proposed method (left), and a visualization of the LiDAR pointcloud and the map (right).

5. Conclusions

In this work, a novel localization method for autonomous vehicles is proposed. It consists of a modified version of AMCL where GNSS data are integrated. As it is shown in Section 4, this method combines the strengths of the two combined localization methods, without compromising accuracy at any scenario, urban and non-urban. The results prove a good performance and a smooth transition from using GNSS data when the map is featureless to using LiDAR and map data when GNSS localization is not accurate, improving localization when both sources are available. The improvements depend on the environment but the proposed method always gives better results, or, in the worst case, the same results. It also gives a low-cost solution for different enviroments using low-cost sensors such as one layer LiDAR or low accuracy GNSS receiver compared to systems that use High-cost RTK or differential GNSS receivers and multiple layer LiDARs, making possible the generalization of these applications. The method is tested and evaluated with a well-known database (KITTI) which is usually used to evaluate autonomous vehicles perception and localization algorithms and with a real platform, improving its performance thanks to a better localization. Besides, the source code of the method is published so it can be tested and improved by anyone.

Author Contributions: M.Á.d.M. wrote the manuscript and carried out the investigation and methodology, F.G. reviewed the manuscript and the methodology and J.M.A. reviewed the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: Research supported by the Spanish Government through the CICYT projects (TRA2016-78886-C3-1-R and RTI2018-096036-B-C21), Universidad Carlos III of Madrid through (PEAVAUTO-CM-UC3M) and the Comunidad de Madrid through SEGVAUTO-4.0-CM (P2018/EMT-4362).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kuutti, S.; Fallah, S.; Katsaros, K.; Dianati, M.; Mccullough, F.; Mouzakitis, A. A survey of the state-of-the-art localization techniques and their potentials for autonomous vehicle applications. *IEEE Internet Things J.* **2018**, *5*, 829–846. [CrossRef]
2. Karaim, M.; Elsheikh, M.; Noureldin, A. GNSS Error Sources. In *Multifunctional Operation and Application of GPS*; IntechOpen: London, UK, 2018.
3. Huang, D.; Xiong, Y.; Yuan, L. *Global Positioning System (GPS)-Theory and Practice;* Xi'an Jiaotong University Press: Chengdu, Chain, 2006; pp. 71–73.
4. Fox, D.; Burgard, W.; Dellaert, F.; Thrun, S. Monte carlo localization: Efficient position estimation for mobile robots. *AAAI/IAAI* **1999**, *1999*, 1–7.
5. Gutmann, J.S.; Fox, D. An experimental comparison of localization methods continued. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Lausanne, Switzerland, 30 September–4 October 2002; Volume 1, pp. 454–459.
6. Martí, E.D.; Martín, D.; García, J.; De la Escalera, A.; Molina, J.M.; Armingol, J.M. Context-aided sensor fusion for enhanced urban navigation. *Sensors* **2012**, *12*, 16802–16837. [CrossRef] [PubMed]
7. Obst, M.; Bauer, S.; Reisdorf, P.; Wanielik, G. Multipath detection with 3D digital maps for robust multi-constellation GNSS/INS vehicle localization in urban areas. In Proceedings of the 2012 IEEE Intelligent Vehicles Symposium, Alcala de Henares, Spain, 3–7 June 2012; pp. 184–190.
8. Qin, H.; Xue, X.; Yang, Q. GNSS multipath estimation and mitigation based on particle filter. *IET Radar Sonar Nav.* **2019**, *13*, 1588–1596. [CrossRef]
9. Ahn, K.; Kang, Y. A Particle Filter Localization Method Using 2D Laser Sensor Measurements and Road Features for Autonomous Vehicle. *Int. J. Rail Transp.* **2019**, *2019*, 1–11. [CrossRef]
10. Sefati, M.; Daum, M.; Sondermann, B.; Kreisköther, K.D.; Kampker, A. Improving vehicle localization using semantic and pole-like landmarks. In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA, 11–14 June 2017; pp. 13–19.
11. Zhou, X.; Su, Z.; Huang, D.; Zhang, H.; Cheng, T.; Wu, J. Robust global localization by using global visual features and range finders data. In Proceedings of the 2018 IEEE International Conference on Robotics and Biomimetics (ROBIO), Kuala Lumpur, Malaysia, 12–15 December 2018, pp. 218–223.
12. Fentaw, H.W.; Kim, T.H. Indoor localization using magnetic field anomalies and inertial measurement units based on Monte Carlo localization. In Proceedings of the 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN), Milan, Italy, 4–7 July 2017; pp. 33–37.
13. Pfaff, P.; Burgard, W.; Fox, D. Robust monte-carlo localization using adaptive likelihood models. In Proceedings of the 1st European Robotics Symposium (EUROS-06), Palermo, Italy, 16–18 March 2006; pp. 181–194.
14. Ueda, R.; Arai, T.; Sakamoto, K.; Kikuchi, T.; Kamiya, S. Expansion resetting for recovery from fatal error in monte carlo localization-comparison with sensor resetting methods. In Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No. 04CH37566), Sendai, Japan, 28 September–2 October 2004; pp. 2481–2486.
15. Goto, H.; Ueda, R.; Hayashibara, Y. Resetting Method using GNSS in LIDAR-based Probabilistic Self-localization. In Proceedings of the 2018 IEEE International Conference on Robotics and Biomimetics (ROBIO), Kuala Lumpur, Malaysia, 12–15 December 2018; pp. 1113–1118.
16. Seow, Y.; Miyagusuku, R.; Yamashita, A.; Asama, H. Detecting and solving the kidnapped robot problem using laser range finder and wifi signal. In Proceedings of the 2017 IEEE international conference on real-time computing and robotics (RCAR), Okinawa, Japan, 14–18 July 2017; pp. 303–308.
17. Lenser, S.; Veloso, M. Sensor resetting localization for poorly modelled mobile robots. In Proceedings of the 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065), San Francisco, CA, USA, 24–28 April 2000; pp. 1225–1232.
18. Hentschel, M.; Wulf, O.; Wagner, B. A GPS and laser-based localization for urban and non-urban outdoor environments. In Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, 22–26 September 2008; pp. 149–154.
19. Perea, D.; Morell, A.; Toledo, J.; Acosta, L. GNSS integration in the localization system of an autonomous vehicle based on Particle Weighting. *IEEE Sens. J.* **2019**, *20*, 3314–3323. [CrossRef]

20. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets Robotics: The KITTI Dataset. *Int. J. Rob. Res.* **2013**, *32*, 1231–1237. [CrossRef]
21. Quigley, M.; Conley, K.; Gerkey, B.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; Ng, A.Y. ROS: an open-source Robot Operating System. In Proceedings of the ICRA Workshop on Open Source Software, Kobe, Japan, 12–17 May 2009.
22. Moore, T.; Stouch, D. A Generalized Extended Kalman Filter Implementation for the Robot Operating System. In Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13), Padua, Italy, 15–18 July 2014; pp. 335–348.
23. Wan, E.A.; Van Der Merwe, R. The unscented Kalman filter for nonlinear estimation. In Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373), Lake Louise, Alberta, Canada, 4 October 2000; pp. 153–158.
24. Bresson, G.; Alsayed, Z.; Yu, L.; Glaser, S. Simultaneous localization and mapping: A survey of current trends in autonomous driving. *IEEE Trans. Intell. Veh.* **2017**, *2*, 194–220. [CrossRef]
25. Balasuriya, B.; Chathuranga, B.; Jayasundara, B.; Napagoda, N.; Kumarawadu, S.; Chandima, D.; Jayasekara, A. Outdoor robot navigation using Gmapping based SLAM algorithm. In Proceedings of the 2016 Moratuwa Engineering Research Conference (MERCon), Moratuwa, Sri Lanka, 5–6 April 2016; pp. 403–408.
26. Marti, E.; de Miguel, M.A.; Garcia, F.; Perez, J. A Review of Sensor Technologies for Perception in Automated Driving. *IEEE Intell. Transp. Syst. Mag.* **2019**, *11*, 94–108. [CrossRef]
27. Milošević, M.; Stefanović, M. Performance Loss Due to Atmospheric Noise and Noisy Carrier Reference Signal in Qpsk Communication Systems. *Elektron. ir Elektrotechnika* **2005**, *58*, 5–9.
28. Thrun, S.; Burgard, W.; Fox, D. *Probabilistic Robotics*; MIT Press: Cambridge, UK, 2000.
29. Burgard, W.; Stachniss, C.; Bennewitz, M.; Grisetti, G.; Arras, K. Introduction to Mobile Robotics. Available online: http://domino.informatik.uni-freiburg.de/teaching/ss13/robotics/slides/12-pf-mcl.pdf (accessed on 1 June 2020).
30. de Miguel, M.A.; Moreno, F.M.; Garcia, F.; Armingol, J.M. Autonomous vehicle architecture for high automation. In Proceedings of the International Conference on Computer Aided Systems Theory, Las Palmas de Gran Canaria, Spain, 17–22 February 2019; pp. 145–152.