

# Multi Sensor Fusion for Navigation and Mapping in Autonomous Vehicles: Accurate Localization in Urban Environments

L. Qingqing<sup>1,2</sup>, J. Peña Queraltá<sup>2</sup>, T. Nguyen Gia<sup>2</sup>, Z. Zou<sup>1</sup> and T. Westerlund<sup>2</sup>

<sup>1</sup> School of Information Science and Technology, Fudan University, China

<sup>2</sup> Department of Future Technologies, University of Turku, Finland

Emails: {qingqingli16, zhuo}@fudan.edu.cn,

{jopequ, tunggi, toweve}@utu.fi,

**Abstract**—The combination of data from multiple sensors, also known as sensor fusion or data fusion, is a key aspect in the design of autonomous robots. In particular, algorithms able to accommodate sensor fusion techniques enable increased accuracy, and are more resilient against the malfunction of individual sensors. The development of algorithms for autonomous navigation, mapping and localization have seen big advancements over the past two decades. Nonetheless, challenges remain in developing robust solutions for accurate localization in dense urban environments, where the so called last-mile delivery occurs. In these scenarios, local motion estimation is combined with the matching of real-time data with a detailed pre-built map. In this paper, we utilize data gathered with an autonomous delivery robot to compare different sensor fusion techniques and evaluate which are the algorithms providing the highest accuracy depending on the environment. The techniques we analyze and propose in this paper utilize 3D lidar data, inertial data, GNSS data and wheel encoder readings. We show how lidar scan matching combined with other sensor data can be used to increase the accuracy of the robot localization and, in consequence, its navigation. Moreover, we propose a strategy to reduce the impact on navigation performance when a change in the environment renders map data invalid or part of the available map is corrupted.

**Index Terms**—SLAM; Sensor fusion; Navigation; Localization; Mapping; Urban navigation; ROS; PCL; 3D Lidar; LOAM; Last-mile delivery; Autonomous vehicles; Self-driving cars;

## I. INTRODUCTION

Accurate mapping and localization is the cornerstone of self-driving cars. In open roads or highways, lane-following can partly replace localization while still allowing for autonomous operation [1]. Nonetheless, in dense urban environments accurate localization is a paramount aspect of a robot's autonomous operation [2]. Smaller pathways and more dynamic environments pose significant technical challenges [3]. In addition, the robot mission often adds accuracy requirements, such as in autonomous post delivery [4]. Multiple sensors can be utilized to facilitate autonomous navigation and operation. Among those, visual data provides more semantic and qualitative information [5], [6], lidar measurements are more accurate and are able to accurately describe objects from a geometric point of view [7].

The past decade has seen a boost in the development of autonomous vehicles for civilian use. Google started the development of its self-driving technology for cars in 2009 [8], and since then a myriad of industry leaders [9], [10], startups [11], and academic researchers [12] have joined the race in

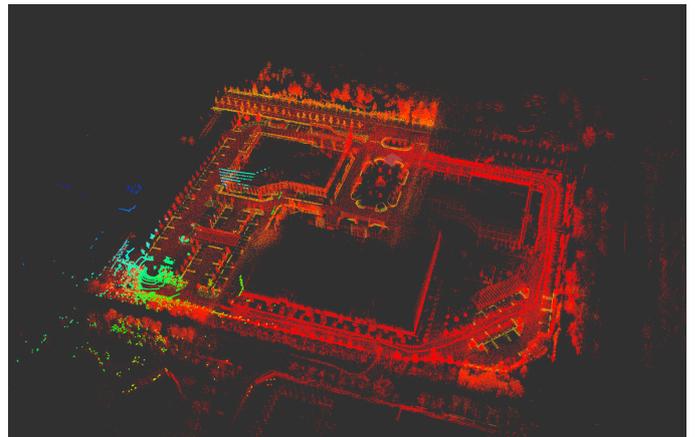


Fig. 1. Illustration of the matching process between the pre-acquired map (red) and current lidar scan (green-blue).

the technology sector, a race to make everything autonomous. In any mobile robot or vehicle, SLAM algorithms are an essential and crucial aspect of autonomous navigation [2], [13].

Autonomous robots or self-driving cars will potentially disrupt the logistics industry worldwide [14]. Autonomous trucks or autonomous cargo vessels are already in advanced stages or development and might be seen in operation within the next five or ten years [15]. However, both technological and legal challenges remain within the so called "last-mile" delivery [16]. Last-mile refers to the last step in the delivery of goods from a local logistics or supply center to the clients' door. In this paper, we utilize data gathered using a small delivery robot from Jingdong, one of the top two e-commerce platform and B2C online retailer in China.

The development of simultaneous localization and mapping (SLAM) algorithms has seen a rapid evolution over the past two decades [17], [18]. In SLAM algorithms, information from a wide range of sensors can be used to map the environment and localize the vehicle or robot in real time. These include inertial measurement units, monocular or binocular cameras, GNSS sensors, lidars, ultrasonic sensors or radars, or wheel encoders [19]. Detailed 3D maps in the form of point clouds can be generated, for instance, from 3D lidars or with stereo vision [7]. We focus on the study and comparison of different localization methods for a small delivery robot in dense urban environments. In these scenarios, an existing map of the

operating environment is obtained in advance and either pre-loaded or accessible by the autonomous robot. The map is used in order to obtain more accurate localization by matching each scan with a certain area of the map in real-time [2], [20].

The local motion of a robot or vehicle can be estimated directly by integrating data from inertial measurement units, including accelerometers, gyroscopes and compasses. Alternatively, different odometry methods can be applied based on non-inertial sensors. Visual odometry algorithms utilize feature extraction and tracking, while lidar-based odometry uses mostly geometric information [17], [18]. Inertial measurement units can be easily combined with wheel encoders. Differential GNSS measurements also provide accurate local motion estimation [21]. Global localization can be estimated either with GNSS data, or by comparing sensor data with predefined maps or information gathered a priori. For instance, different methods exist to match a lidar scan with section of a 3D point cloud that defines a map of the operational area [2], [20]. Over the past decades, researchers from both industry and academia have been exploring the utilization of these methods and their combinations to obtain accurate mapping and localization. More concretely, scholars often refer to the combination of different sensor data as sensor fusion or data fusion. In this paper, we compare different techniques and provide arguments on the best sensor fusion techniques for a small delivery robot for last-mile delivery.

The algorithms, analysis and results presented in this paper were mostly developed during the JD Digital (JDD) Globalization Challenge Competition in "Multi-sensor fusion localization for autonomous driving". The challenge was a global competition, with 4 classification divisions depending on the geographical location of the team. Our team ranked first in the US division semi-final and classified for the 24h global final in Beijing, China, where the 4 semi-final winners competed for the first prize. The available sensor data during the competition was GNSS and gyroscope data, wheel odometry and the output from a 16-channel 3D lidar. A map of the area was given as a 3D point cloud. Multiple datasets exist to test and benchmark different localization algorithms. However, the most accurate algorithms are obtained through fine tuning and parameters specific for the dataset, with different parameters being potentially necessary to achieve the optimal accuracy in a different sensor or environment setup [22]. Therefore, in this paper we have utilized the data provided as part of the JDD competition as it was gathered from the sensors on-board the vehicle in order to compare a variety of methods. This ensures that our algorithms can be implemented on the same robot without a significant impact to performance.

The main contributions of this work are the following: (i) to analyze and compare different approaches for vehicle localization estimation, and the definition of a sensor fusion approach for accurate localization in urban environments; and (ii) the introduction of a strategy for rebuilding an area of a local map when data is corrupted or the environment has undergone significant modifications.

The remainder of this paper is organized as follows: Section II introduces related work in localization and odometry techniques based on data available from the on-board sensors. Section III describes the dataset used in the paper, delves

into the possibilities of sensor fusion approaches based on the available information, and provides a strategy to operate in areas where the available map is either outdated or corrupted. Then, section IV shows experimental results on the localization accuracy for four different sensor fusion approaches. Finally, Section V concludes the work and outlines directions for future work.

## II. RELATED WORK

Autonomous navigation through 3D mapping with lidars has been an increasingly popular technology over the last decade, as lidars can provide high accuracy range measurements when compared to other sensors. Zhang *et al.* proposed a method for lidar odometry and mapping. The authors approached the odometry problem by extracting lidar point cloud features from each sweep. Then, the transformation between two consecutive sweeps can be estimated. In this setup, the lidar is utilized as an odometry sensor [17]. For 3D mapping, the most important problem is accurate position estimation, and 3d lidar data for odometry still can produce accumulated error after long distance walk. However, the accumulated error is much lower when compared to camera-based odometry (especially in unfavourable light conditions) or integration of inertial data from accelerometers/gyroscopes. State-of-the art solutions combine lidar and monochrome camera sensors as visual-lidar odometry to improve the performance of ego-motion estimation [18].

In SLAM algorithms, localization and mapping are done concurrently in real time. However, in order to achieve localization with centimeter accuracy in an urban environment, map matching techniques have emerged over the past few years. Currently, one of the most widely used approaches for 3D point cloud matching is Normal Distributions Transform (NDT) matching [23]. Introduced by Bibel *et al.*, NDT has the advantages of no requiring explicit assignments of relationships between features or subsets of points, and the analytic formulation of the algorithms. The former aspect increases the robustness of the algorithm, while the latter reduces the computational cost and accuracy of the implementation.

Multiple improvements of the NDT algorithm have been proposed. Gonzalez Prieto *et al.* presented DENDT, an algorithm for 3D-NDT scan matching with Differential Evolution [20]. The authors utilize a differential evolution (DE) algorithm in order to improve the optimization process for finding the solution of the NDT method. Akai *et al.* presented a robust localization method that uses intensity readings from the lidar data in order to detect road markings and use them for matching consecutive scans [2]. While the method is able to provide very high accuracy in some environments, the existence of road markers significantly impacts the performance and therefore this method cannot be used in all situations. Wen *et al.* recently analyzed the performance of different NDT-based SLAM algorithms in multiple scenarios in Hong Kong [24]. A valuable conclusion from this work is that the best performance was achieved in areas with more sparse point clouds and nominal traffic conditions, while the performance decreased in dense urban areas. In this work, we study how can we combine different sensor information in order to improve localization performance in an urban area.

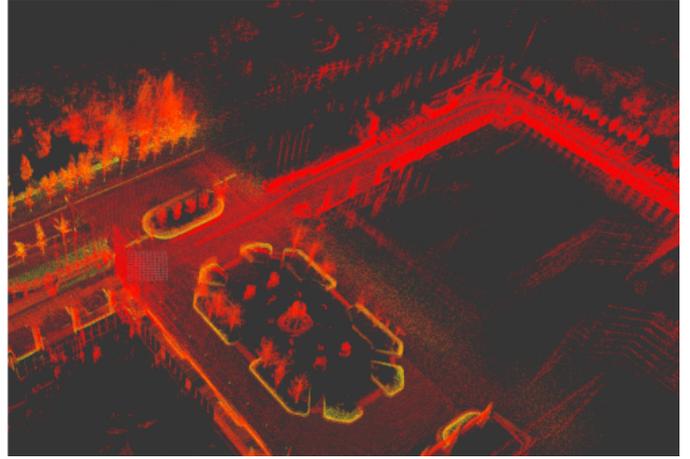
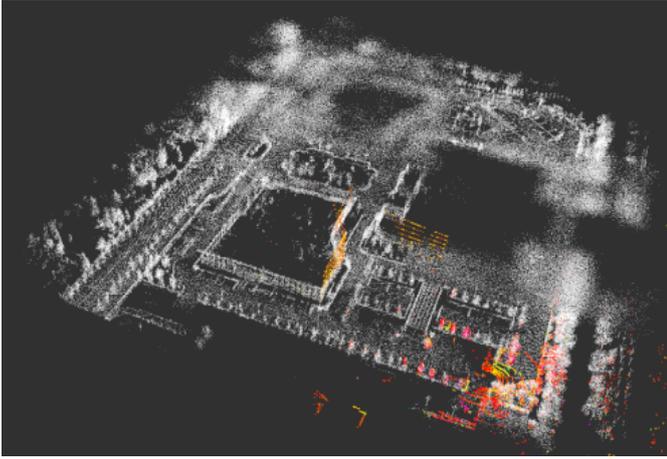


Fig. 2. On the left, a map with noise added in some areas to simulate corrupted data. On the right, one of the two corrupted sections of the map has been restored using GNSS, IMU and lidar odometry.

### III. LOCALIZATION AND MAPPING

In this section, we describe the dataset that we have utilized and the different localization approaches. We also introduce a strategy for situations where the existing map data is corrupted, outdated, or part of the data is missing.

#### A. Dataset

The data utilized in this paper was provided as part of the JD Discovery Global Digitalization Challenge from December 2018 to January 2019. The data was gathered using JD’s autonomous last-mile delivery robot. The data includes: (1) GNSS directional and positional data referenced in the World Geodetic System (WGS84) format; (2) lidar data as a 3D point cloud; (3) raw accelerometer and gyroscope data; and (4) wheel speed meter. Ground truth data is provided as well. The output of the 3D lidar is given at 10 Hz, IMU data is acquired at 100 Hz and GNSS data is updated at 5 Hz. In addition, a map of the objective operation area is given. The map represented as a point cloud is shown in red in fig. 1 on page 1. The dataset contains sensor data recorded in an 800-second long closed loop movement.

In order to both read and process data, ROS has been utilized. ROS (Robot Operating system) is an open source operating system for robots, which provides a publish-subscribe communication framework that allows for rapid development of distributed robotic systems [25]. ROS provides algorithm reuse support for robot research and development, as well as abstraction of data models for easier integration of different modules. PCL (Point Cloud Library) is a cross-platform open source C++ library, which implements common algorithms and data structures of point clouds [26]. It can realize point cloud acquisition, filtering, segmentation, registration, retrieval, feature extraction, recognition, tracking, surface reconstruction, visualization and so on. If OpenCV is the crystallization of 2D visual data acquisition and processing, PCL has the same position in the 3D geometrical data domain.

#### B. Localization methods

Based on the available sensor data, we have utilized five different approaches to estimate the vehicle’s localization. In

each approach, we use a different combination of sensors and describe how the robot position is calculated based on their data.

##### *GNSS-based localization*

One of the most traditional methods for outdoor robot localization is to use a global navigation satellite system. In this case, data from multiple satellite constellations was available and used for increase accuracy. GNSS data error are mostly caused by the atmospheric conditions and multi-path interference. The effect of the environment in a larger scale and the atmospheric conditions can be minimized using differential GNSS readings, and assuming that the real-time error is equivalent to the error obtained in a near known location with which the system is synchronized. However, in this work we have not relied on differential GNSS.

##### *GNSS+IMU localization*

We can easily combine GNSS data with inertial data, including both accelerometer and accuracy. As differential GNSS has not been implemented in this case, instead, the results labelled as “IMU” utilize the IMU readings for local motion estimation, and the GNSS reading for an initial global estimation and estimations when the robot movement is almost zero for a prolonged period of time.

##### *Lidar odometry (LOAM)*

Zhang *et al.* introduced lidar odometry as an alternative to the more classical visual odometry techniques [17]. As with many odometry approaches, features are extracted from data and compared within consecutive frames, or scans in the case of a lidar. Features extracted from lidar data are usually based on geometrical aspects. These include corners and surfaces, for instance. Because lidars are able to provide high accuracy distance measurements even for objects far away from the sensors, lidar-based odometry is able to provide higher accuracy than visual-based odometry in open space situations with clearly differentiated objects. An implementation based on Zhang’s algorithm has been used in this case.

##### *NDT-based localization (NDT+)*

The NDT algorithm is a kind of registration algorithm that uses the existing high-precision point cloud map and real-time 3D lidar point cloud data to achieve high-precision localization.

NDT algorithm does not directly compare the distance between points in point clouds map and points in lidar point clouds. First, the NDT algorithm will transform the point cloud map into the normal distribution in three-dimensional.

If a variable  $X$  is normal distribution  $X \sim (\mu, \delta)$ , then it can be described as:

$$f(x) = \frac{1}{\delta\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\delta^2}} \quad (1)$$

where  $\mu$  is the mean of the variable distribution and  $\delta^2$  is the variance. For a multivariate normal distribution, its probability density function can be expressed as:

$$f(\vec{x}) = \frac{1}{(2\pi)^{\frac{D}{2}} \sqrt{|\Sigma|}} e^{-\frac{(\vec{x} - \vec{\mu})^T \Sigma^{-1} (\vec{x} - \vec{\mu})}{2}} \quad (2)$$

Where  $\vec{x}$  represents the mean vector and  $\Sigma$  represents the covariance matrix.

The first step of the NDT algorithm is to divide the point cloud into a 3D grid coordinate. For each cell, the probability distribution function(PDF) is calculated based on the points distribution density in the grid.

$$\vec{\mu} = \frac{1}{m} \sum_{k=1}^m \vec{y}_k \quad (3)$$

$$\Sigma = \frac{1}{m} \sum_{k=1}^m m(\vec{y}_k - \vec{\mu})(\vec{y}_k - \vec{\mu})^T \quad (4)$$

Where  $\vec{y}_k = 1, 2, 3, m$  denotes all lidar points in a grid. Then the PDF can be expressed as:

$$f(\vec{x}) = \frac{1}{(2\pi)^{\frac{3}{2}} \sqrt{|\Sigma|}} e^{-\frac{(\vec{x} - \vec{\mu})^T \Sigma^{-1} (\vec{x} - \vec{\mu})}{2}} \quad (5)$$

We use the normal distribution to represent the discrete points of each grid. Each probability density function can be considered as an approximation of a local surface. It not only describes the location of the surface in space but also contains information about the direction and smoothness of the surface.

After calculated the PDF of each grid, then our goal is to find the best transformation. The lidar point cloud set is  $X = \vec{x}_1, \vec{x}_2, \dots, \vec{x}_n$ , and the parameter of transformation is  $\vec{p}$ . The spatial transformation function  $T(\vec{p}, \vec{x}_k)$  represents using transformation  $\vec{p}$  to move point  $\vec{x}_k$ , combined with the previous calculated state density function(the PDF of each grid), so the best transformation  $\vec{p}$  should be the transformation of maximum likelihood function:

$$Likelihood : \theta = \prod_{k=1}^n f(T(\vec{p}, \vec{x}_k))$$

And the maximum likelihood is equivalent to minimum negative logarithmic likelihood:

$$-\log \theta = -\sum_{k=1}^n \log f(T(\vec{p}, \vec{x}_k))$$

The task now becomes to minimize the negative logarithmic likelihood by using an optimization algorithm to adjust the transformation parameter  $\vec{p}$ . we can use the Newton method to optimize the parameters.

The main problem of the NDT approach is its stability when used standalone. As indicated by the authors of previous works, NDT alone has the disadvantage of being unstable depending on the scenario [24]. Therefore, we utilize GNSS data for setting the initial position as well as resetting the NDT localization method when a sudden change in position or orientation estimation is detected. In the results, we refer to this method as NDT+, and a close implementation to the one provided in existing NDT method has been used [24].

### NDT+IMU localization (NDT++)

The final localization utilized in our experiments consisted on integrating the IMU data into the NDT+ method described above that uses lidar and GNSS data. With this approach, we have been able to eradicate the instabilities of the NDT+ method and increase its accuracy.

The algorithm workflow is as follows: first, on system start-up or reset, GNSS data is used in order to obtain an initial estimation of the robot's location. This estimation can be utilized in order to reduce the area of the map in which the NDT matching will be looked for. Second, when the robot starts moving, an unscented Kalman filter that uses IMU data as input serves as an estimation between lidar scans. The Kalman filter output is then feeded to the NDT algorithm for scan matching with the predefined map. The GNSS data is still used to avoid instabilities, even though we have not detected any in the dataset utilized in this paper.

The NDT+ and NDT++ approaches have an additional benefit over the lidar odometry method. In autonomous robots moving in an urban environment, it is essential to react on time to obstacles and to have localization information as frequently as possible. A lidar-only approach has the disadvantage of receiving sensor updates at only 10Hz in this case. With IMU readings having a refresh rate of 100Hz, the IMU can be utilized to obtain local movement estimation between lidar scan matches using the NDT approach. This minimizes the possibilities of instabilities in the NDT algorithm as the matching possibilities are reduced and the goal of the algorithm partly shifts from coarse localization to increasing the accuracy of IMU-based movement estimation.

### C. Corrupted Map Reconstruction

In an urban scenario, it is impractical to propose a localization method that has a high dependency on the existence of an accurate map of the operational area without a strategy for operating in case the map data is corrupted or outdated. In fig. 2on the previous page, we show the map of the operation area (on the left, in black and white), with two areas where the data has either been removed completely or noise has been added to render the NDT algorithm unusable. When the robot approaches these areas, we are able to detect them by monitoring the difference between the NDT localization and GNSS and IMU positioning. When part of the map cannot be matched with current scans, we utilize lidar odometry and mapping in order to rebuild the corrupted or missing data.

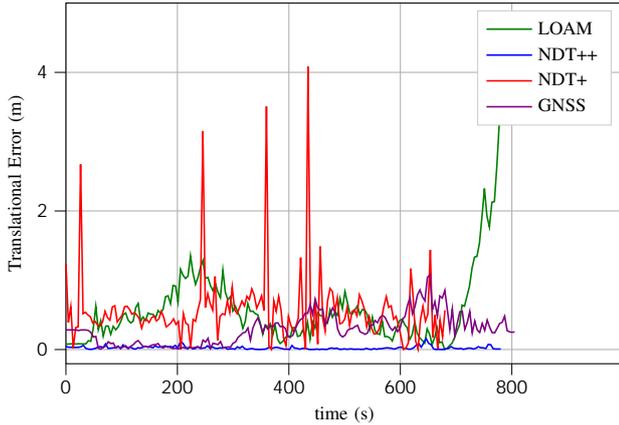


Fig. 3. Translational errors of the proposed approaches over time.

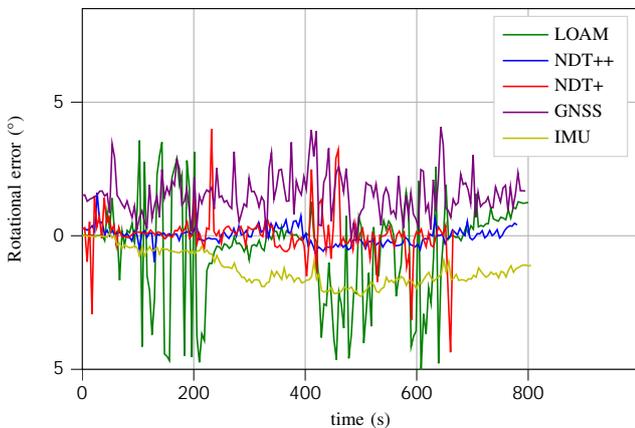


Fig. 4. Rotational errors of the proposed approaches over time.

The result of this process is shown on the right side of fig. 2 on page 3, where one of the corrupted map areas has been restored in real-time while the robot was travelling through it using lidar odometry and mapping. Even though it is not visible in the image, there is a relatively small mismatch in the map in the area where the robot emerges again into a mapped environment.

#### IV. EXPERIMENT AND RESULTS

We have applied the five approaches proposed to the given dataset. The results showing the translational and rotational localization and orientation error are shown in fig. 3 and fig. 4, respectively. In these figures, the NDT++ method shows a stable and very small error though time, both in position and rotation estimation. The NDT+ without taking into account inertial data shows a larger error but, more importantly, shows several instabilities that are corrected from the GNSS data.

In order to be able to compare in more detail the different methods, and to see whether there exist some background error or drifting, we show the variability of the localization error though two sets of boxplots in fig. 5 and fig. 6 on the following page. The specific values are also listed in Table I. We have omitted the location errors of standalone IMU motion estimation as the error is significantly higher than the proposed approaches. However, inertial data is still valuable for local movement estimation and for orientation estimation.

TABLE I  
LOCALIZATION ERROR MEAN AND STADARD DEVIATION

	$\mu_{rot.}$	$\sigma_{rot.}$	$\mu_x$	$\sigma_x$	$\mu_y$	$\sigma_y$
<b>GNSS</b>	1.52	0.79	-0.46	0.22	$10^{-4}$	0.18
<b>IMU</b>	-1.24	0.62	n/a	n/a	N/A	N/A
<b>LOAM</b>	-0.50	1.88	0.40	0.49	0.10	0.49
<b>NDT+</b>	0.03	0.87	-0.02	1.51	-0.05	1.13
<b>NDT++</b>	$10^{-3}$	0.31	-0.01	0.10	-0.05	0.10

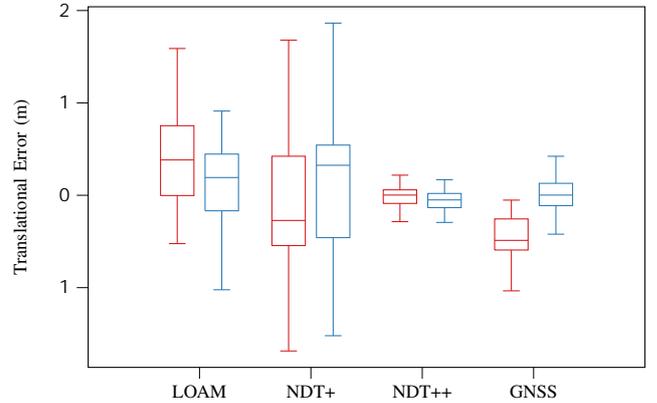


Fig. 5. Boxplots of translational errors for the different approaches: x-error (red) and y-error (blue).

The translational errors are shown in fig. 5, where the red boxes show the error in the  $x$  coordinate, and the blue boxes refer to the  $y$  coordinate error. We have separated the error because in some cases the error mean differs between them. That is the case of the GNSS data, which due to atmospheric or environmental conditions shows a steady negative drift in the  $x$  direction. If consistent through large periods of time, it can be assumed that it is due to a sensing error in the device itself, or to environment conditions such a specific multi-path occurring in the operating area. Therefore, this value can be utilized to decrease the sensing error in real time during operation. In order to have a deeper understanding of the distribution of the GNSS error, we show the histogram of the three errors (two translational, one rotational) in fig. 7 on the following page. Only the error in the  $y$  direction has a mean of 0, while the distribution of the  $x$  error is symmetrical and narrow. Therefore, it is possible to fix the drift while keeping the same variance for both components of the translational error. In the case of the rotational error, it is more complex to correct even though the distribution is still symmetric.

From fig. 5 and fig. 6 on the following page we can see that the most stable methods are NDT++ and GNSS, with the NDT+ method providing accurate results for rotation estimation. However, the NDT+ is highly unstable for position estimation, with the highest variance of all presented approaches. In position estimation, all approaches have a relatively small error after 800 seconds of movement, except for the LOAM method, which error drifts away from 0 towards the end of the available data set. Similarly, the IMU constantly drifts in terms of orientation estimation but it provides a more stable measurement than LOAM, HDL+ or GNSS.

In summary, lidar scan matching with a 3D map provides the highest accuracy for localization, both in terms of position

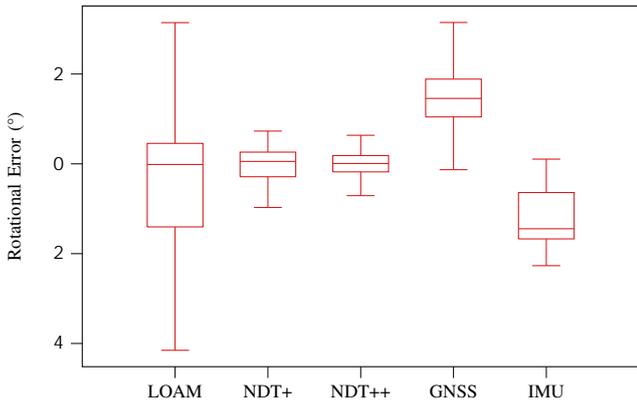


Fig. 6. Boxplots of rotational errors for the different approaches in degrees.

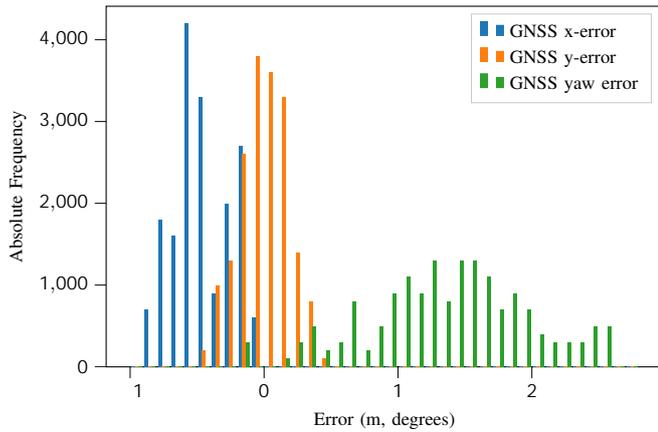


Fig. 7. Distribution of GNSS errors.

and orientation. Nonetheless, it is essential to take into account other sensor data in order to implement a more robust approach that is less prone to instabilities and depends less on the operational environment. GNSS and inertial data are essential for increasing the localization accuracy but also for minimizing the possibilities of unexpected behaviour in the algorithm.

## V. CONCLUSION

Accurate localization in dense urban areas is paramount in order to solve the autonomous last-mile delivery problem. Nonetheless, it still presents important challenges. We have explored the possibilities for localization in a city environment using 3D lidar data complemented with GNSS and inertial data using a delivery robot from JD. We have shown the accuracy of different approaches, assuming that a map of the operation area is given in the form of a point cloud. In addition, we have presented a strategy for situations where the map might be corrupted or the scenario might have undergone significant changes that rendered the map outdated. We have shown that 3D scan matching is the best approach for localization when properly complemented with IMU data within an unscented Kalman filter, and GNSS data. In future work, we will explore the possibilities of integrating visual data for odometry, as well as using the lidar odometry together with the inertial data within the Kalman filter in order to increase the NDT localization accuracy.

## ACKNOWLEDGMENT

This work has been supported by NFSC grant No. 61876039, and the Shanghai Platform for Neuromorphic and AI Chip (NeuHeilium).

## REFERENCES

- [1] R. W. Wolcott. Visual localization within lidar maps for automated urban driving. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 176–183, Sep. 2014.
- [2] N. Akai *et al.* Robust localization using 3d ndt scan matching with experimentally determined uncertainty and road marker matching. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, June 2017.
- [3] J. Levinson *et al.* Map-based precision vehicle localization in urban environments. In *Robotics: Science and Systems*, volume 4, page 1. Citeseer, 2007.
- [4] A. Heinla *et al.* Method and system for autonomous or semi-autonomous delivery, August 16 2018. US Patent App. 15/948,974.
- [5] S. Se *et al.* Vision-based global localization and mapping for mobile robots. *IEEE Transactions on robotics*, 21(3):364–375, 2005.
- [6] E. Garcia-Fidalgo *et al.* Vision-based topological mapping and localization methods: A survey. *Robotics and Autonomous Systems*, 64, 2015.
- [7] C. Premebida *et al.* High-resolution lidar-based depth mapping using bilateral filter. In *2016 IEEE 19th international conference on intelligent transportation systems (ITSC)*, pages 2469–2474. IEEE, 2016.
- [8] M. John. Google cars drive themselves. *Traffic {WWW Document}*. *The New York Times*. URL <http://www.nytimes.com/2010/10/10/science/10google.html>, 2010.
- [9] M. Bojarski *et al.* End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.
- [10] M. Harris. Documents confirm apple is building self-driving car. *The Guardian*, 14, 2015.
- [11] N. A. Greenblatt. Self-driving cars and the law. *IEEE spectrum*, 53(2):46–51, 2016.
- [12] S. Kato *et al.* An open approach to autonomous vehicles. *IEEE Micro*, 35(6):60–68, 2015.
- [13] W. Zhang. Lidar-based road and road-edge detection. In *2010 IEEE Intelligent Vehicles Symposium*, pages 845–848, June 2010.
- [14] E. Hofmann. Industry 4.0 and the current status as well as future prospects on logistics. *Computers in Industry*, 89:23–34, 2017.
- [15] H. Flämig. Autonomous vehicles and autonomous driving in freight transport. In *Autonomous driving*, pages 365–385. Springer, 2016.
- [16] N. Boysen *et al.* Scheduling last-mile deliveries with truck-based autonomous robots. *European Journal of Operational Research*, 271(3):1085–1099, 2018.
- [17] J. Zhang *et al.* Loam: Lidar odometry and mapping in real-time. In *Robotics: Science and Systems*, 2014.
- [18] J. Zhang *et al.* Visual-lidar odometry and mapping: Low-drift, robust, and fast. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015.
- [19] H. Cho *et al.* A multi-sensor fusion system for moving object detection and tracking in urban driving environments. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014.
- [20] P. G. Prieto *et al.* Dendt: 3d-ndt scan matching with differential evolution. In *2017 25th Mediterranean Conference on Control and Automation (MED)*, pages 719–724, July 2017.
- [21] C. Rizos *et al.* Precise point positioning: is the era of differential gnss positioning drawing to an end? 2012.
- [22] The KITTI Vision Benchmark Suite. Visual odometry and slam evaluation, 2012-2019.
- [23] P. Biber *et al.* The normal distributions transform: A new approach to laser scan matching. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, volume 3, pages 2743–2748. IEEE, 2003.
- [24] W. Wen *et al.* Performance analysis of ndt-based graph slam for autonomous vehicle in diverse typical driving scenarios of hong kong. *Sensors*, 18(11):3928, 2018.
- [25] M. Quigley *et al.* Ros: an open-source robot operating system. In *ICRA workshop on open source software*. Kobe, Japan, 2009.
- [26] R. B. Rusu *et al.* 3D is here: Point cloud library (PCL). In *2011 IEEE International Conference on Robotics and Automation*, May 2011.