A Virtual Indoor Localization Testbed for Wireless Sensor Networks

Simon Schmitt, Heiko Will, Thomas Hillebrandt and Marcel Kyas
Freie Universität Berlin
AG Computer Systems & Telematics
Berlin, Germany
{simon.schmitt, heiko.will, thomas.hillebrandt, marcel.kyas}@fu-berlin.de

Abstract—We present a novel, easy to use virtual testbed enabling researchers to evaluate their localization algorithms based on distance measurements in indoor environments. We provide precise ground truth information collected by our previously presented reference system, based on a mobile robot, in combination with range measurements from a Wireless Sensor Network (WSN) in multiple buildings. The user can define a virtual experiment using our datasets over the web. This approach separates the process of designing a robust indoor localization algorithm from the need for testing it on actual hardware in different scenarios.

I. INTRODUCTION

In the field of indoor localization research, simulation is a common approach to get information about the performance of different algorithms. Due to the complexity of modeling a real-world environment in high detail, simulations can often not provide enough insight into real-world behavior. If appropriate hardware components are available, an analysis in real-world scenarios is possible but often very time consuming, since ground truth data is needed to properly evaluate the localization algorithm. We presented a reference system [1] which is based on a remote controlled mobile robot and capable of performing independent localization in a 2D indoor environment with an average error of 6.7 cm. We showed that it is easily feasible to record both, ground truth data and data from a system under test, which is carried by the reference system. The data has to be merged and can be analyzed Afterwards or even during an experiment.

The accuracy of an indoor localization system does not only depend on the localization algorithms, but also highly on several environmental influences. For example, the position of anchors and the node to be localized is much more important than the error of the measured distances [2]. Also structural and dynamic properties like people moving around, or open and closed doors have big influence on the localization accuracy. Because of that, a single experiment has only limited significance; so multiple experiments in the same environment are needed. To simplify these complex processes, our virtual testbed allows the user to repeatedly "move" through several different buildings gathering distance data and ground truth positions from a huge database of collected real-world experiment data.

To build such a database, a robot has to traverse a certain environment (e.g. room, hallway) repeatedly. The anchor configuration has to be changed after each traversal and the recorded data of both systems has to be synchronized and transferred to a database capable of handling position-based information.

To access this data a user interface has to allow the user to specify a virtual path through the environment. The user shall then be able to download a virtual recording containing the ground truth data of the chosen path and either a subset or all measurements from the system under test along that path.

We present such a virtual testbed, based on data acquired by our previously presented reference system [1] and, as a system under test, a modified version of the Modular Sensor Board (MSB) A2 [3] node which is equipped with a Nanotron nanoPAN 5375 [4] transceiver. This hardware enables the sensor nodes to measure inter-node ranges using time-of-flight in the 2.4 GHz frequency band.

We provide a web interface which is directly accessible over the Internet with a standard web browser, no additional software installation is required.

II. SYSTEM OVERVIEW

As shown in Fig. 1, the virtual testbed consists of a reference system that carries a system under test, a data fusion component, a database back end, and a user front end. The position of the reference system is transferred to the data fusion component, along with the raw range measurements of the system under test. The data fusion component interpolates the positions of the system under test and stores these reference positions and the raw data of the system under test in the database. The user can then perform virtual test runs using the user front end.

A. Reference System

We previously presented a reference system for indoor localization testbeds [1]. A low-cost TurtleBot [5] robot carried the system under test while directed remotely through hallways in our office-like building. It consists of a Roomba cleaning robot [6] and a laptop that is mounted on top of the Roomba. We use the Robot Operating System (ROS) [7] and its already provided localization components to estimate the robot’s position relative to a coordinate system defined by a pre-drawn floor plan. For localization, odometry data from the Roomba is combined with visual distance measurements.
taken by a Kinect depth sensor [8] mounted on the robot. An implementation of Adaptive Monte-Carlo Localization [9] is used for localization. The use of a rack allows for mounting an additional mobile component of a system under test. The use of open-source software assures a high customizability for hard- and software. We implemented a middleware that allows us to collect data produced by a system under test as well as the ground truth positions from the robot itself. We are able to continuously calculate the robot’s position with an average position estimation error of 6.7 cm. The system has been evaluated by driving along a grid of known positions while performing self-localization.

To achieve better odometry data we acquired a Q.bo robot [10]. This is a commercial robot capable of robust indoor localization given a pre-drawn 2D map of the environment. It uses the same software components as the previous reference system. Instead of a Microsoft Kinect camera, it uses a topt-mounted Asus Xtion PRO LIVE 3D camera [11] which provides the same data output. It has bigger wheels which allow the traversal of higher door sills. We collect the ground truth data for the virtual testbed using this Q.bo robot as the reference system. Due to its similarities to the TurtleBot, the localization error is comparable. Another benefit of the Q.bo is its robust casing.

The reference system is able to traverse and localize in an environment using a pre-drawn 2D accurate floor plan. The floor plan does not have to contain movable objects like tables or chairs. But it is advised to remove objects like chairs since the reference system cannot traverse that space and would thereby produce a lot of gaps in the virtual testbed.

We use a simple Dijkstra implementation to produce a path fragment of roughly 2 m in our environment, respecting movable objects not in the pre-drawn floor plan. The algorithm searches positions not already traversed while considering the current angle of the robot and the space already traversed earlier. Space near already traversed space is rated higher, so that the algorithm produces a tight path, without criss-crossing the environment. Path fragments are spaced with a distance of approximately 20 cm. When the current target position is reached, another path fragment is computed. While collecting data, the robot traverses its environment with a maximum speed of 10 cm/s, if only driving forward. While turning, slightly more data is acquired on the turning position, depending on the turning angle. The ground truth position is acquired with 10 Hz, so that reference positions with a distance of approximately one centimeter can be transferred to the data fusion component.

B. System Under Test

As the system under test we use our Wireless Sensor Network (WSN) nodes [3]. The nodes consist of a modified version of the Modular Sensor Board (MSB) A2 which is equipped with a Nanotron nanoPAN 5375 [4] transceiver. This hardware enables the sensor nodes to measure inter-node ranges using time-of-flight in the 2.4 GHz frequency band. One mobile node is attached to the reference system. This node tries to perform range measurements using all predefined anchors. To correctly model the real-world behavior, the failure of a range measurement is also stored in the database. When a range measurement was taken, the software running on the Q.bo robot attaches a timestamp to it and transfers it to the data fusion component.

Since we only have a limited amount of nodes available, we cannot fully equip an environment with anchors. Therefore we virtually increase the anchor count by letting the reference system repeat a traversal of the environment using a new anchor placement.

C. Data Fusion

We merge reference data and data from the system under test by using unique timestamps on the robot. The collected information from both systems are stored in two queues, ordered in time. When a position update from the reference system is received all range measurements from the mobile sensor node currently stored in the queue are iterated. The reference position of a range measurement is computed using linear interpolation between two subsequent ground truth positions, defining the time interval surrounding the range measurement. However, the maximum distance between these positions is normally limited by one centimeter due to the design of the reference system. Since the expected localization error of the WSN is significantly higher, linear interpolation suffices and can benefit the integrity of the data if reference positions are delayed for some reason. When such two ground truth positions are available, the range measurement is removed from its queue and sent along with the interpolated reference position to the database back end.

D. Database Back End

A PostgreSQL database installed on a separate server is used to store the data received via a HTTP REST interface from the data fusion component on the reference system. We also store the actual hardware identifier of the sensor node, since it might be the case that single nodes produce false measurements, e.g. due to defects or wrong calibration. This can later be evaluated.
using the filled database. We use PostGIS, a spatial database extender, to efficiently run queries that output only those range measurements that were received on a certain position.

We designed the database to store data received from a time-of-flight based measurement system like ours, but it can easily be adapted to accommodate the needs of other systems, e.g., storing RSSI values, data from barometers or other fingerprinting data.

E. User Front End

The database back end can be accessed via a website. A scalable map of the building in which the runs were recorded is presented to the user. The user can define a virtual experiment by drawing any path on the map. Then, the user has to choose between anchor positions that were available during the recording. The ground truth positions on that path and the range measurements from the WSN received while traversing these ground truth positions are then available for download. Due to the low speed of the reference system, multiple range measurements with the same reference position are stored in the database. To get a more realistic virtual experiment, the user can download only a random subset of measurements.

To find gaps in the recordings, a heat map shows the distribution of ground truth data. If a gap is too big (e.g., more than 20 cm wide), the reference system is manually directed along these gaps, to ensure a mostly homogeneous data distribution.

III. Conclusion

We presented a virtual testbed, which can be used by researchers over the Internet in order to evaluate various localization algorithms. It separates the process of designing a robust algorithm from the need of testing it on actual hardware in a real-world environment. Collecting ground truth information is not necessary any more. The testbed provides raw time-of-flight measurement data from our WSN with accurate ground truth data collected in a real-world scenario. Any path within our dataset can be defined and the recorded data along that path can be downloaded. The huge amount of provided data allows the statistical evaluation of many different localization algorithms, without the need of running several complex and time consuming real-world experiments.

REFERENCES