2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

An Inexpensive Method for Evaluating the Localization Performance of a Mobile Robot Navigation System

Harsha Kikkeri and Gershon Parent and Mihai Jalobeanu and Stan Birchfield
Microsoft Robotics, Redmond, WA 98052
{harshk, gershonp, mihaijal, stanleyb}@microsoft.com

Abstract— **We propose a method for evaluating the localization accuracy of an indoor navigation system in arbitrarily large environments. Instead of using externally mounted sensors, as required by most ground-truth systems, our approach involves mounting only landmarks consisting of distinct patterns printed on inexpensive foam boards. A pose estimation algorithm computes the pose of the robot with respect to the landmark using the image obtained by an on-board camera. We demonstrate that such an approach is capable of providing accurate estimates of a mobile robot's position and orientation with respect to the landmarks in arbitrarily-sized environments over arbitrarily-long trials. Furthermore, because the approach involves minimal outfitting of the environment, we show that only a small amount of setup time is needed to apply the method to a new environment. Experiments involving a state-of-the-art navigation system demonstrate the ability of the method to facilitate accurate localization measurements over arbitrarily long periods of time.**

I. Introduction

Quantifying the performance of a robotic system is important for the purpose of being able to objectively compare different systems, measure the progress of the research endeavor, and determine whether a system is qualified for a given application. While performance metrics and benchmarks are common in other engineering domains as well as other branches of robotics (e.g., industrial manipulators), autonomous navigation lacks such established metrics and methods. Even though there are multiple reasons why quantifying the performance of a navigation system is difficult, ranging from theoretical (what exactly should be measured?) to practical (what should the environment look like?), one remaining barrier is the lack of a simple, inexpensive, and non-intrusive method of accurately and independently determining the actual position of the robot (ground truth).

For such a method to scale to real environments of many types and sizes (factory floors, office buildings, warehouses, stores, etc.), the simplicity of the system is key. First, the setup time required should be at most linear in the size of the environment. Second, there should be nothing about the technology used by the system that prevents its deployment in large environments. Furthermore, the system must be inexpensive to allow wide deployment in many environments, including environments of various sizes. Finally, the system needs to be non-intrusive to enable deployment in real environments without disrupting any daily activity taking place. Being non-intrusive is critical in enabling performance testing over long periods of time (e.g., days or weeks).

We propose a method for measuring the localization performance of a navigation system that satisfies these requirements. The approach requires very simple outfitting of an environment, namely placing foam boards with printed patterns at specified waypoints on the ceiling along with an upward facing camera on the robot. For practical reasons we focus on indoor environments and wheeled robots, though the method could be adapted to remove these restrictions. The localization performance is determined by measuring the accuracy and precision of position and orientation at the sequence of waypoints chosen in a task- and environment-specific manner. Because the system is both inexpensive and easy to set up, it scales well and is thus applicable to real indoor environments of virtually any size and any type, as well as to essentially any mobile robot system. Equally important, the limited impact of the foam boards on the indoor environment enables measuring localization accuracy over arbitrarily long running times. Moreover, automatically determining the location of the robot over long periods of time is critical for measuring other aspects of navigation performance, such as speed, safety, and reliability. We describe the procedure for using this method and evaluate its accuracy, as well as validate its use by measuring the performance of a state-of-the-art navigation system.

II. Previous Work

Over the past decade the robotics community has begun to devote more attention to developing means of comparing and benchmarking algorithms. This shift mirrors a trend in computer vision that began with work in face detection and stereo [8], [9]. For the most part, the focus has been on validating system components like path planning and tracking, as in the case in [2], where an external camera system covering an area of 25 m^2 provided ground truth information with an accuracy of 3 cm. In the case of SLAM, one of the earliest attempts is the Radish dataset,[1] which is an open collection of data shared by researchers all over the world for the purpose of evaluating and comparing algorithms. The repository enables qualitative assessment of SLAM results (e.g., visual aspect of the resulting map), but the lack of ground-truth information prevents a quantitative assessment. Similar collections can be found online [10],[2,3]

[1] http://radish.sourceforge.net/
[2] http://www.robots.ox.ac.uk/NewCollegeData
[3] http://www.informatik.uni-freiburg.de/~stachnis/datasets.html

978-1-4799-3684-7/14/$31.00 ©2014 IEEE

as well as open-source algorithms to aid comparison;[4] and some attempts have been made at defining methods for comparison [1].

More recently, significant effort has been put toward creating SLAM benchmark datasets with ground truth information, enabling quantitative evaluation of algorithms. The Rawseeds project [3], [5][5] contains multisensor datasets, along with benchmark problems and solutions; ground truth was obtained via a set of fixed lasers and cameras throughout the environment, all calibrated together. The vision system achieves an accuracy of 112 mm and -0.8 degrees, while the laser system achieves an accuracy of 20 mm and 0.2 degrees. The SLAM benchmark at the University of Freiburg [4], [7] proposes to measure the performance of SLAM systems using "relative relations" between poses, basing all computation on the corrected trajectory of the robot without regard to a global coordinate system or environment model. The authors validate the benchmark on Radish datasets augmented with ground truth generated by manually aligning sensor data. More recently, Sturm et al. [12], [11] present the Technical University of Munich's RGB-D SLAM dataset and benchmark,[6] for which ground truth was obtained using a motion capture system with an accuracy of 10 mm and 0.5 degrees.

All of these benchmarks rely on pre-recorded data paired with ground-truth information. Since data is recorded with particular equipment in a particular environment, it is difficult to compare solutions that use a different sensor suite or are optimized for a different environment. Additionally, while existing benchmarks cover the problem of creating maps (SLAM), they cannot be used to compare navigation performance. Furthermore, the methods used to gather ground truth are time and cost prohibitive to reproduce in other labs, and they do not scale well to larger environments. These are some of the problems that our method is designed to address.

Another set of efforts revolves around qualitative evaluation of indoor navigation systems through contests, such as RoboCupRescue [6][7] or RoboCupHome [15],[8] where the focus is on achieving a particular task such as finding victims or successfully navigating to a set of learned locations. While there has been a series of efforts to define benchmarks for robotics in general and navigation in particular, such as the EURON Benchmarking Initiative[9] and its related workshops,[10] the Robot Standards and Reference Architectures (RoSta) project,[11] and the Performance Metrics for Intelligent Systems (PerMIS) workshop[12] organized by the National Institute of Standards and Technology (NIST), these efforts have not yet generated a definitive set of performance metrics and benchmarks for navigation. Two years ago NIST conducted navigation performance tests at the Alabama Robot Technology Park (RTP) near Huntsville, Alabama,[13] but correspondence with the organizers has revealed that while the project was successful at comparing a number of systems qualitatively, no results have been shared with the community yet.

The remotely accessible Teleworkbench [13] has a similar aim as ours, namely to facilitate comparison of mobile robot algorithms. In that work, small robots navigate a small 13 m^2 space, and robot positions are determined by reading barcodes mounted on the top of the robots using overhead cameras. Like the other approaches, this system does not have the ability to scale to large environments without significant investment in infrastructure.

Tong and Barfoot [14] describe a system which involves placing large retroreflective markers on walls throughout an environment. A 360-degree 3D laser is used to detect the markers, whose positions are determined via a SLAM algorithm. Like our approach, this retroreflective system is scalable to large environments and, once set up, can be used to continuously measure the robot's location. The system achieves relative accuracy on the order of tens of millimeters in position and half a degree in orientation. To our knowledge, this is the only other published work for benchmarking navigation systems on large scales in a quantitative way. Our approach is arguably simpler to deploy (since it does not require a scanning 3D laser) and is less likely to be affected by occlusion, at the expense of not computing absolute position throughout the space.

III. Approach

A. Overview

Our method requires placing *landmarks*, namely printed patterns on foam boards, throughout the environment. A calibration procedure is performed to determine the tilt of the floor underneath each landmark. A separate calibration procedure is used to determine the transformation between the coordinate system of the camera (which is mounted on the robot) and the coordinate system of the robot. After these steps, the system is in place and can be used to evaluate the performance of a robot navigation system automatically and for indefinite periods of time. We now describe these steps in more detail.

B. Landmarks

The landmarks are foam boards on which are printed a pattern that allows a pose estimation algorithm to determine the pose of the camera with respect to the landmark. Possible choices for patterns include various AR (augmented reality) tags or otherwise unique configurations of monochrome or color shapes to facilitate detection and pose estimation. We use a custom pattern consisting of a black-and-white checkerboard with a grid of 14 squares by 10 squares; the inner 4×4

[4] http://openslam.org
[5] http://www.rawseeds.org
[6] http://vision.in.tum.de/data/datasets/rgbd-dataset
[7] http://wiki.robocup.org/wiki/Robot_League
[8] http://wiki.robocup.org/wiki/Home_League
[9] http://www.euron.org/activities/benchmarks
[10] http://www.robot.uji.es/EURON/en
[11] http://www.robot-standards.eu
[12] http://www.nist.gov/el/isd/ks/permis.cfm
[13] https://sites.google.com/site/templearra/Welcome

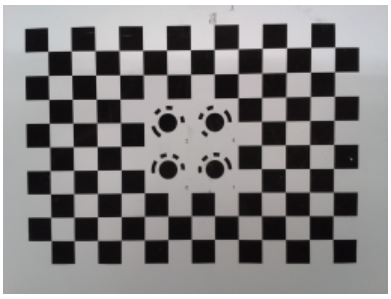

Fig. 1. The checkerboard pattern used for pose estimation.

area replaced by four circles centered with respect to each of the four quadrants of this inner area, as shown in Figure 1. Beyond the radius of each circle are four arcs of varying length and spacing, enabling the circles to be distinguished from one another. This pattern is easily detectable, even when a significant portion of the checkerboard is occluded from the field of view, and it enables the orientation of the pattern to be determined. The landmarks are oriented horizontally and attached to the ceiling (or otherwise mounted on a stand) so that they can be viewed by an upward-facing camera.

C. Estimating camera pose

When at least one quadrant of the landmark is visible in the current field-of-view of the camera, pose estimation software computes the 6 degree-of-freedom pose of the camera in 3D space. The software that we use [16] returns the landmark's pose in camera-centric coordinates, which we then convert to find the camera's pose in landmark-centric coordinates. We ignore the yaw and pitch angles, since they are too noisy to be of any use for our application. Although the robot drives on a relatively flat floor, the z value (along the optical axis) is needed, because the ceiling height is not guaranteed to be constant throughout an environment. As a result, we retain the x, y, and z coordinates, along with the roll θ of the camera about its optical axis, which is directly related to the orientation of the robot.

D. Calibration

There are four coordinate systems that are relevant to our problem. The *image coordinate system* is placed at the upper-left of the image plane and oriented along the image axes. The *camera coordinate system* is centered at the focal point and oriented in the same direction as the image coordinate system. The *robot coordinate system* is centered with the robot and aligned with the robot driving direction. The *landmark coordinate system* is centered on the landmark and aligned with the checkerboard. Except for the image coordinate system, which is measured in pixels, all measurements are in millimeters.

Calibrating the image-to-camera transformation involves estimating the internal camera parameters, which is done using the well-known algorithm of Zhang [16]. Calibrating the camera-to-robot transformation involves estimating 6 parameters: the tilt ϕ_c of the camera with respect to the floor normal, the azimuth θ_c of this camera tilt plane (i.e., the plane containing the gravity vector and the optical axis) with respect to the forward direction of the robot, the tilt ϕ_f of the floor (in the immediate vicinity below the landmark) with respect to gravity, the azimuth θ_f of the floor tilt plane (i.e., the plane containing the gravity vector and the normal vector to the floor) with respect to the positive x axis of the landmark coordinate system, and the lateral offset of the focal point from the robot center, expressed in polar coordinates as d_{rc} and θ_{rc}.

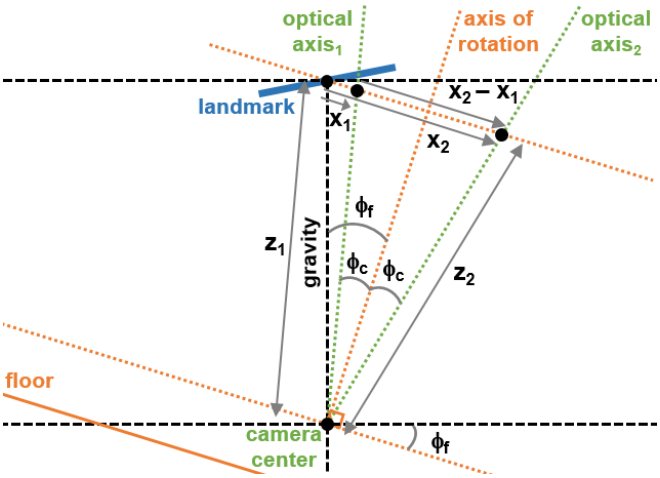

Fig. 2. Side view of a simplified camera-to-robot calibration process. Considering only the vertical plane shown, the pose estimation software yields the pose (x_1, z_1) of the camera with respect to the landmark. Then the robot is rotated 180 degrees, and the software yields the pose (x_2, z_2). From these measurements, the quantities of interest, namely ϕ_c and ϕ_f can be computed. Note that the image plane (not shown) is perpendicular to the optical axis and is not (in general) parallel to the floor.

To determine these 6 parameters, we place the camera directly under the landmark using a self-leveling line laser so that the vertical laser beams (we ignore the horizontal beams) intersect the center of the image and the center of the landmark, as shown in Figure 2. We then rotate the robot by fixed increments, being careful to ensure that the axis of rotation passes through the focal point (i.e., camera center). The figure shows a side view of the geometry of the system as the camera is rotated by 180 degrees, sliced by the xz-plane. The pose estimation software measures the (x, y, z) coordinates of the landmark with respect to the camera both before and after the rotation, leading to (x_1, y_1, z_1) and (x_2, y_2, z_2). From the figure, the camera tilt is given by $\phi_c = \sin^{-1}((x_2 - x_1)/2\bar{z})$, and the floor tilt is given by $\phi_f = \sin^{-1}((x_1 + (x_2 - x_1)/2)/\bar{z}) = \sin^{-1}((x_2 + x_1)/2\bar{z})$, where $\bar{z} = (z_1 + z_2)/2$. In the case of zero floor tilt we have $\phi_f = 0$, $x_1 = -x_2$, and $\phi_c = \sin^{-1}(x_2/\bar{z})$. Notice that, since the pose estimation software yields the pose of the landmark in the camera coordinate system, the angle of the landmark does not matter in any case.

When the camera tilt is non-zero, rotating the robot in the manner described causes the optical axis to trace the shape of a cone about the axis of rotation, which is assumed to be perpendicular to the floor. Therefore, as the robot is rotated, the (x, y) coordinates returned by the pose

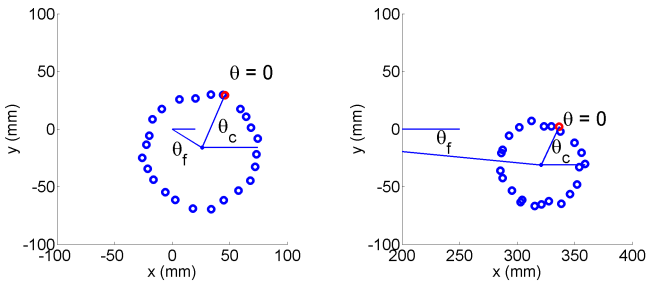

Fig. 3. Noisy circles traced by the optical axis underneath two different landmarks. The robot was rotated in 15-degree increments, leading to 24 data points. The red data point is the one for which the robot was aligned with the landmark at $\theta = 0$.

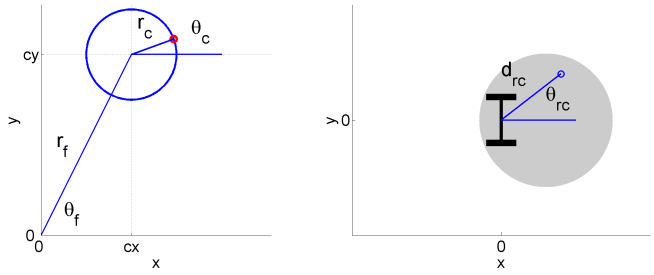

Fig. 4. Left: The circle traced by the optical axis yields 4 of the 6 calibration parameters. Right: The 2 additional parameters capture the robot-to-camera offset. (The gray circle is the robot, the black I represents the wheels and axle, and the blue circle is the camera.)

estimation software trace a circle, two examples of which are shown in Figure 3.

If we let (x_i, y_i, z_i), $i = 1, \ldots, n$, be the readings taken as the robot is rotated, then the center of the circle is estimated as the mean of the coordinates: $(c_x, c_y) = \frac{1}{n}\sum_{i=1}^{n}(x_i, y_i)$; the radius of the circle is estimated by the average Euclidean distance to the center: $r_c = \frac{1}{n}\sum_{i=1}^{n}\sqrt{(x_i - c_x)^2 + (y_i - c_y)^2}$; and the distance from the circle center to the origin is given by $r_f = \sqrt{c_x^2 + c_y^2}$. We use $n = 4$ readings, rotating the robot to each of 0°, 90°, 180°, and 270° positions. In theory only two readings 180° apart are sufficient, but more readings provide additional robustness to noise.

Note from Figure 2 that the radius of the circle is also given by $r_c = (x_2 - x_1)/2$, and the distance from the landmark center to the circle center is $r_f = (x_2 + x_1)/2$. Therefore, the camera tilt is given by $\phi_c = \sin^{-1}(r_c/\bar{z})$, the floor tilt is $\phi_f = \sin^{-1}(r_f/\bar{z})$, and the floor azimuth is $\theta_f = \text{atan2}(c_y, c_x)$, where $\bar{z} = \frac{1}{n}\sum_{i=1}^{n} z_i$, as shown in Figure 4. Assuming that (x_1, y_1) corresponds to $\theta = 0$ (robot is aligned with the landmark), then the camera azimuth is $\theta_c = \text{atan2}(y_1 - c_y, x_1 - c_x)$, and the camera-to-robot offset is measured manually on the ground to determine the distance d_{rc} between the robot and camera centers, as well as the angle θ_{rc}.

Once the system has been calibrated, the tilt-corrected 2D camera pose in the ground plane (that is, corrected for floor tilt) for a given 3D pose (x, y, z) and θ is given by

$$\theta' = \theta \tag{1}$$

$$x' = x - z\sin\phi_c\cos(\theta_c + \theta) - z\sin\phi_f\cos\theta_f \tag{2}$$

$$y' = y - \underbrace{z\sin\phi_c}_{r_c}\sin(\theta_c + \theta) - \underbrace{z\sin\phi_f}_{r_f}\sin\theta_f. \tag{3}$$

Note that the heading is not affected by tilt. The robot pose is then calculated as a simple pose transformation:

$$\theta_r = \theta' + \theta_{rc} \tag{4}$$

$$x_r = x' + d_{rc}\cos\theta_{rc} \tag{5}$$

$$y_r = y' + d_{rc}\sin\theta_{rc}. \tag{6}$$

E. Measuring performance

After mounting the landmarks, mounting the upward-facing camera to the robot, and calibrating, the system is ready to be used to measure the performance of a navigation system automatically and for arbitrary lengths of time. First the robot is driven around the environment to build a map. The nature of the map is completely irrelevant to the proposed evaluation method, thus enabling different types of approaches to be compared; the robot is free to generate a 2D metric map, 3D metric map, topological map, topo-geometric map, or otherwise, or to operate purely reactively. The robot can be driven manually, or it can autonomously explore. Whenever the robot is under a landmark for the first time, the user clicks a button (or something similar) to remember the location. This button press causes the system being evaluated to store the location of the robot with respect to the map being constructed. Simultaneously, the button press triggers the pose estimation algorithm to analyze the image from the upward-facing camera and, with calibration, to determine the robot's pose with respect to the landmark. Note that this approach completely decouples the internal map representation of the system being evaluated from the evaluator.

Once the map has been built, the system being evaluated contains a set of locations with respect to its internal map, while the evaluator also contains a set of locations with respect to the landmarks. Each landmark has an ID, and this ID is the same in both sets of locations to enable correspondence to be made between the system being evaluated and the evaluator. To evaluate the system, the evaluator generates a sequence of waypoints (IDs), and the robot is commanded to visit these waypoints in sequence. When the robot system determines that it has reached the desired location (waypoint), it notifies the evaluator, which then analyzes the image from the upward-facing camera to determine the robot's pose with respect to the landmark. The discrepancy between the pose of the robot during map-building and the pose of the robot during evaluation yields an indication of the accuracy of the navigation system.

IV. Experimental Results

We divide the experimental results into two parts. First we evaluate the accuracy of the pose estimation system, then we

validate the proposed method by measuring the performance of a state-of-the-art navigation system.[14]

A. Evaluating accuracy of pose estimation

To evaluate the accuracy of the pose estimation, we attached a 14×10 inch checkerboard pattern printed on a 470×336 mm foam board to the ceiling of our lab. Each square of the checkerboard was 33.6×33.6 mm. The landmark was placed 2200 mm above a CNC machine (Fireball Meteor,[15] $1320 \times 640 \times 127$ mm) capable of xyz translation with a precision of 0.0635 mm. To the carriage of the CNC machine we attached a Microsoft LifeCam Cinema camera, facing upward, with a 73 degree field of view, capable of capturing images at a resolution of 1280×720. The camera could see the landmark from a distance of 300 to 3000 mm; at the distance of 2200 mm the checkerboard occupied a space of 300×220 pixels in the image and could be seen within an area approximately 1500×1500 mm on the ground. To reduce effects due to the low contrast caused by ceiling lights, we turned off the automatic gain control and reduced the exposure level.

The camera was rotated on the CNC until the pose estimation software indicated zero degree pan (and independently validated by making sure that the checkerboard edges were parallel to the image edge in the captured image). The long edge of the CNC was aligned with the long end of the checkerboard. To align the CNC axis with the landmark axis, the camera was moved along one direction and the CNC was rotated until the pose estimation software showed change in only one direction.

The CNC was calibrated by moving it to the home location. The CNC carriage was then moved in the x and y directions until the pose estimation software indicated the (x, y, θ) offset to be $(0, 0, 0)$. This was considered as the origin and the CNC carriage position was noted. Once calibrated, the CNC carriage was moved within the 1320×640 mm area at 5 mm increments. At each position the machine was stopped for 1 to 2 seconds to remove any vibration, then an image was taken by the camera, and the pose estimation software estimated the pose of the camera. Figure 5 shows the results. The average position error was 5 mm ($\sigma = 2$ mm), and the average orientation error was 0.3 degrees ($\sigma = 0.2$ degrees). Within the entire area the position error never exceeded 11 mm, and the orientation error never exceeded 1 degree. In a followup experiment the CNC carriage was moved within the same area at 40 mm increments, but at each position 360 images were taken (at 1 degree intervals) using a dynamixel MX-64 servo mechanism to rotate the camera. The pose was estimated for each image, and the results were consistent.

From perspective projection, it is easy to see that the maximum error due to pixel quantization is

$$\text{quantization error} = \frac{pz}{f}, \tag{7}$$

[14]We use the latest version of Adept MobileRobots' navigation software: Active ARNL Laser Localization Library 1.7.5 for Windows.

[15]http://www.probotix.com/FireBall_Meteor_cnc_router

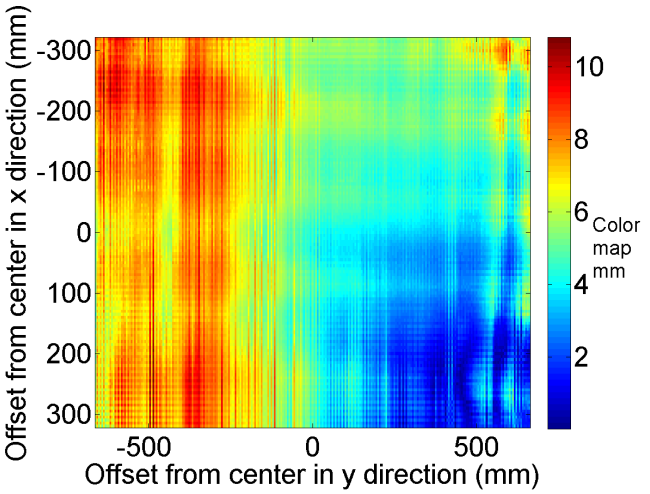

Fig. 5. Euclidean error of the pose estimation algorithm over a 1320×640 mm area, obtained via motions generated by a CNC machine.

Fig. 6. Robot with upward-facing camera attached.

where f is the focal length of the camera (4.8 mm), z is the distance to the landmark (2200 mm), and p is the pixel size (5 μm). Given these values, the error due to pixel quantization is approximately 2 mm, which is consistent with these results.

We then evaluated the accuracy of the system with the camera mounted on the Adept Pioneer 3DX mobile robot shown in Figure 6. After placing 15 landmarks across 2 buildings (described below), we arbitrarily chose one of the landmarks and placed the robot at 20 random positions / orientations underneath it, as shown in Figure 7a. The actual (r_x, r_y, r_θ) of the robot was measured manually on the ground and compared with the values calculated as described above. The resulting Euclidean distance and orientation errors were computed, shown in Figure 8.

At each of these landmarks the robot was then placed at 5 canonical locations (units are mm): $(x, y, \theta) = (0, 0, 0°)$, $(400, 0, 0°)$, $(0, 400, 90°)$, $(-400, 0, 180°)$, and $(0, -400, 270°)$, as shown in Figure 7b. The Euclidean distance and orientation errors were calculated at all the landmarks for these five canonical locations by comparing

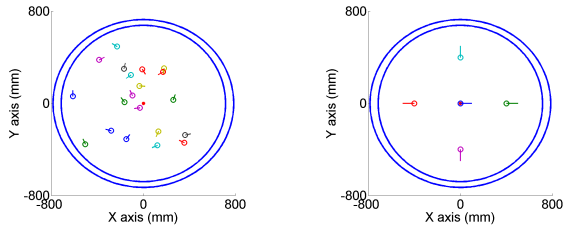

Fig. 7. 20 random (left) and 5 canonical (right) positions / orientations used for evaluating the accuracy of pose estimation.

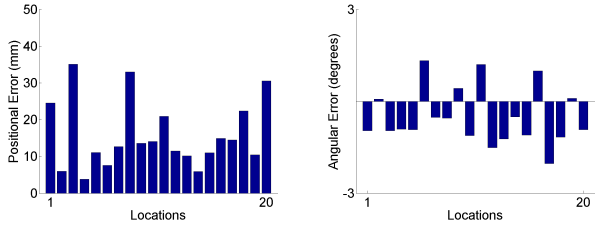

Fig. 8. Position and orientation error for 20 random positions / orientations.

with manually obtained values. The results are shown in Figure 9. Since the results were similar for both experiments, we only report the combined mean position error, 15.2 mm ($\sigma = 9.1$ mm), and mean orientation error, $-0.4°$ ($\sigma = 0.8°$). The maximum errors were 52.9 mm and 2.8°.

Table I compares our landmark-based system with several existing ground-truth systems.[16] Our system yields an average of 15 mm position error, with an average angular error of $-0.4°$, over an area approximately 1.5×1.5 m. The Rawseeds GTvision system [5] yields an order of magnitude more error in position, and about twice as much error in orientation, but over an area nearly two orders of magnitude larger. The Rawseeds GTlaser system [5], the mocap system at TUM [11], and the retroreflective SLAM system [14] yield

[16]Data for Rawseeds, TUM's mocap system, and the retroreflective system are from [5, §7.3], [11, §VI-C], and [14, §V-A], respectively.

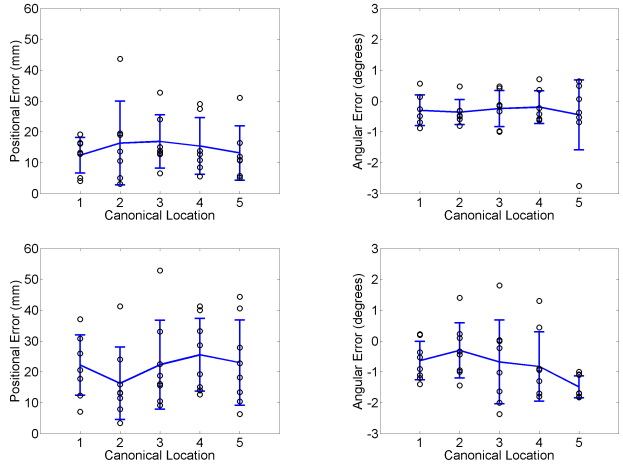

Fig. 9. Position and orientation error for the five canonical positions for Building 99 (top) and Building 115 (bottom). Shown are the error bars (blue) for one standard deviation, along with all the data (black circles).

	dist. (mm)		ang. (deg.)			environment
	mean	s. d.	mean	s. d.	big	size (m)
GTvision	112	90	-0.8	2.2	N	10×14
GTlaser	20	11	**0.2**	1.6	N	10×14
mocap	**10***	–	0.5*	–	N	10×12
retroreflective	21*	14*	0.5*	0.3*	Y	7×5
our system	15	**9**	-0.4	**0.8**	Y	1.5×1.5

TABLE I

COMPARISON OF MEAN AND STANDARD DEVIATION ERROR OF OUR SYSTEM WITH OTHER GROUND-TRUTH SYSTEMS. ONLY OURS AND THE RETROREFLECTIVE SYSTEM SCALE TO LARGER ENVIRONMENTS. THE ASTERISKS INDICATE THAT THE RETROREFLECTIVE SYSTEM ONLY REPORTS ERRORS RELATIVE TO OTHER ROBOT POSITIONS, AND THAT THE NUMBERS FOR THE MOCAP SYSTEM ARE MAXIMUM (NOT MEAN) ERRORS.

errors that are less than ours, also over much larger areas. It is difficult to compare these results, since the purpose of the systems is different: Motion capture and Rawseeds work over a larger area, whereas the accuracy of our system is limited to a fairly narrow field of view. On the other hand our method (as well as the retroreflective system) apply over arbitrarily large environments, whereas the other techniques are generally limited to a single room due to the cost and difficulty of installation and maintenance.

B. Verifying the methodology using an existing robot navigation system

We used an existing robot navigation system to verify the ease of deployment, ease of use, and accuracy of the proposed method. To verify the ease of deployment, we tracked the time and expense involved in setting up the system for use in two environments, Building 115 and Building 99, shown in Figures 10 and 11. Building 115 is approximately 55 m by 23 m consisting of mostly uniform hallways with 10 foot drywall ceilings. The Building 99 space is approximately 57 m by 52 m and contains a combination of open space and hallways with ceilings made of a variety of materials including drywall, acoustic tile, and decorative wood paneling. We installed seven landmarks in Building 99, and eight landmarks in Building 115. Installation of each landmark required about 25 minutes, including surveying the available space, affixing the landmark to the ceiling, and performing the landmark calibration procedure. The robot camera calibration was calculated at the same time as the landmarks. We used a professional printing service to create our foam-core landmarks, which were attached with 3M Damage Free Hanging strips, for a total cost of less than $200 USD. In each case the landmark position was chosen to provide ample coverage of the building. The landmarks in Building 115 were positioned at hallway junctions, while the landmarks in Building 99 were positioned at points of interest around the building (see Figure 12). One of the challenges in Building 99 were locations where flooring changed from carpet to tile, and where floor vents and floor access panels were present, since an uneven floor under the landmark reduces accuracy in position measurement. Such locations

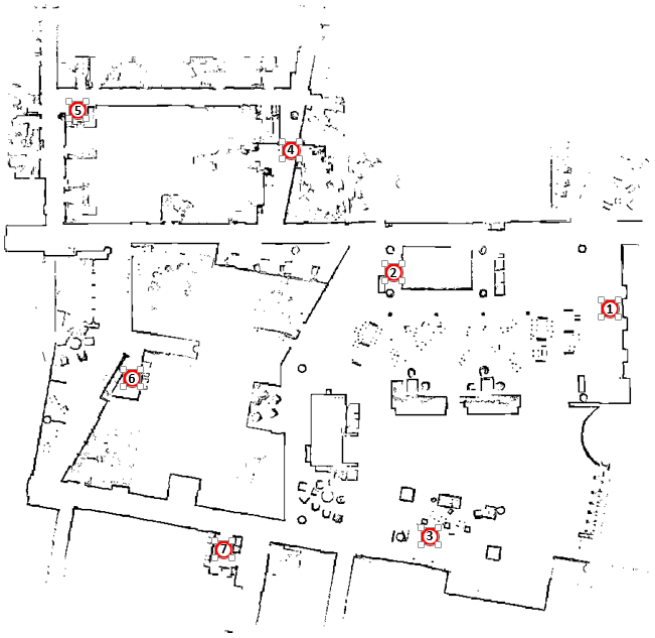

Fig. 10. Map of Building 99. The landmarks were placed at the elevators (1), cafe (2), lounge (3), workshop (4), copy room (5), kitchen (6), and supply room (7).

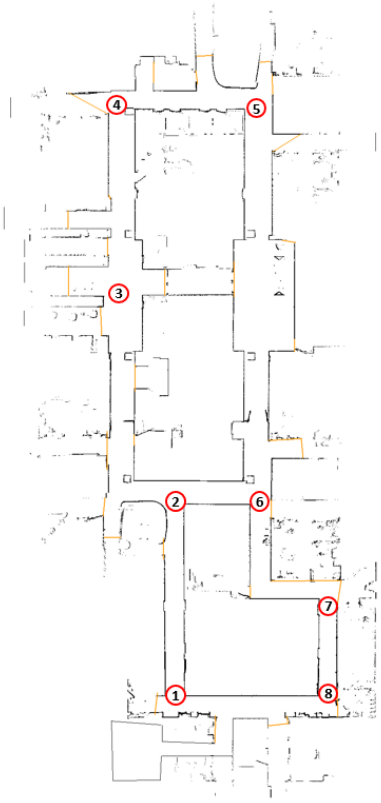

Fig. 11. Map of Building 115, with the 8 landmarks identified.

could still be used to provide a binary result (i.e., whether the robot is under the landmark), but to enable accurate measurement we only chose areas with uniform floor level in the vicinity of the landmark.

To verify the ease of use, Building 115 was used to conduct automated navigation tests over two days during which the robot navigated to a 108-sequence of waypoints (landmarks) in random order. The position measurement system operated entirely autonomously and provided detailed position and orientation error information for the duration of the 8 hours of testing without any manual intervention and could continue to do so as long as the landmarks do not move, and the camera remains calibrated. In fact, these landmarks have been in continuous use for the past several months in our lab, providing daily metrics information for autonomous robot navigation. One advantage of the system is that, once deployed, no further maintenance is needed: There are no cables or power cords, and the landmarks do not interfere with occupants' daily activities.

To verify the accuracy, the Building 99 space was used to conduct automated navigation tests during which the same Adept Pioneer 3DX robot was used to map the space and record goals at each landmark, then navigate a randomly generated 15-waypoint sequence. During mapping, the position and orientation of the robot was recorded by the landmark system, and at the same time its position and orientation was marked manually on the floor using colored duct tape. After the sequence had been travered, the Euclidean distance between the original mapped position and each visited position was measured manually, as well as the difference in orientation. The manual measurement was conducted very carefully, introducing errors no more than a few millimeters. Results are shown in Figure 13, where the close agreement between the manually obtained values and automatically obtained values is evident. These results are consistent with the measurements presented above. Note that the focus of this paper is not on the performance of the navigation system itself, but rather on the ability of our ground truth system to accurately monitor the navigation system.

V. Conclusion

We have presented a method for evaluating the localization performance of a mobile robot navigation system. The method consists of attaching landmarks, namely foam boards on which checkerboard patterns are printed, to the ceiling at various locations around an environment. An upward-facing camera on the robot, along with pose estimation software, is then used to estimate the robot's pose with respect to the landmark. Contrary to previous approaches, the approach is inexpensive, easy to deploy, very low maintenance, and highly accurate. The system enables automatic performance evaluation of arbitrarily sized environments for arbitrary lengths of time. We have performed a thorough evaluation of the system, showing that accuracies on the order of 15 mm and 0.4 degrees can be obtained with proper calibration. Future work will be aimed at using such a method to perform long-term comparison of state-of-the-art navigation systems in a variety of environments.

Fig. 12. 5 of the 7 landmarks used in Building 99. From left to right: cafe, supply room, elevator, kitchen, and workshop.

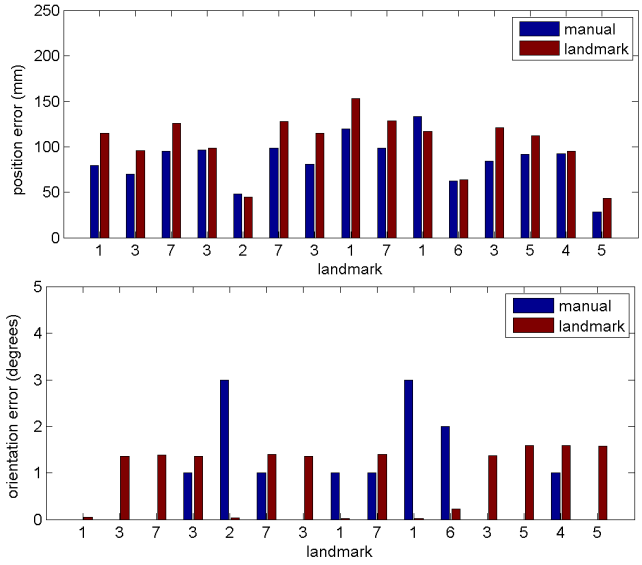

Fig. 13. Position and orientation error for the 15-waypoint sequence. Shown are the absolute errors (with respect to each landmark) determined manually and by using the automatic landmark pose estimation system.

ACKNOWLEDGMENTS

Thanks to Steve Marschner and Cha Zhang who developed the specific checkerboard pattern that we use. Also thanks to Adept for their support in using the Pioneer robot.

REFERENCES

[1] F. Amigoni, S. Gasparini, and M. Gini. Good experimental methodologies for robotic mapping: A proposal. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pages 4176–4181, Apr. 2007.

[2] J. Baltes. A benchmark suite for mobile robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 2, pages 1101–1106, 2000.

[3] A. Bonarini, W. Burgard, G. Fontana, M. Matteucci, D. G. Sorrenti, and J. D. Tardos. RAWSEEDS: Robotics advancement through web-publishing of sensorial and elaborated extensive data sets. In *Proceedings of the IROS Workshop on Benchmarks in Robotics Research*, 2006.

[4] W. Burgard, C. Stachniss, G. Grisetti, B. Steder, R. Kümmerle, C. Dornhege, M. Ruhnke, A. Kleiner, and J. D. Tardós. A comparison of SLAM algorithms based on a graph of relations. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2089–2095, Oct. 2009.

[5] S. Ceriani, G. Fontana, A. Giusti, D. Marzorati, M. Matteucci, D. Migliore, D. Rizzi, D. G. Sorrenti, and P. Taddei. Rawseeds ground truth collection systems for indoor self-localization and mapping. *Autonomous Robots*, 27(4):353–371, 2009.

[6] A. Jacoff, E. Messina, B. A. Weiss, S. Tadokoro, and Y. Nakagawa. Test arenas and performance metrics for urban search and rescue robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 4, pages 3396–3403, 2003.

[7] R. Kümmerle, B. Steder, C. Dornhege, M. Ruhnke, G. Grisetti, C. Stachniss, and A. Kleiner. On measuring the accuracy of SLAM algorithms. *Autonomous Robots*, 27(4):387–407, Nov. 2009.

[8] H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38, 1998.

[9] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1):7–42, 2002.

[10] M. Smith, I. Baldwin, W. Churchill, R. Paul, and P. Newman. The New College vision and laser data set. *International Journal of Robotics Research*, 28(5):595–599, May 2009.

[11] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 573–580, Oct. 2012.

[12] J. Sturm, S. Magnenat, N. Engelhard, F. Pomerleau, F. Colas, W. Burgard, D. Cremers, and R. Siegwart. Towards a benchmark for RGB-D SLAM evaluation. In *Proc. of the RGB-D Workshop on Advanced Reasoning with Depth Cameras at Robotics: Science and Systems (RSS)*, June 2011.

[13] A. Tanoto, J. V. Gómez, N. Mavridis, H. Li, U. Rückert, and S. Garrido. Teletesting: Path planning experimentation and benchmarking in the Teleworkbench. In *European Conference on Mobile Robots (ECMR)*, Sept. 2013.

[14] C. H. Tong and T. D. Barfoot. A self-calibrating 3D ground-truth localization system using retroreflective landmarks. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, May 2011.

[15] T. Wisspeintner, T. van der Zant, L. Iocchi, and S. Schiffer. RoboCupHome: Scientific competition and benchmarking for domestic service robots. *Interaction Studies*, 10(3):392–426, 2009.

[16] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.