2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

An Augmented Reality Spatial Referencing System for Mobile Robots

Sonia Mary Chacko,[1] Armando Granado,[1] Ashwin RajKumar,[1] and Vikram Kapila[1]

***Abstract*— The deployment of a mobile service robot in domestic settings is a challenging task due to the dynamic and unstructured nature of such environments. Successful operation of the robot requires continuous human supervision to update its spatial knowledge about the dynamic environment. Thus, it is essential to develop a human-robot interaction (HRI) strategy that is suitable for novice end users to effortlessly provide task-specific spatial information to the robot. Although several approaches have been developed for this purpose, most of them are not feasible or convenient for use in domestic environments. In response, we have developed an augmented reality (AR) spatial referencing system (SRS), which allows a non-expert user to tag any specific locations on a physical surface to allocate tasks to be performed by the robot at those locations. Specifically, in the AR-SRS, the user provides a spatial reference by creating an AR virtual object with a semantic label. The real-world location of the user-created virtual object is estimated and stored as spatial data along with the user-specified semantic label. We present three different approaches to establish the correspondence between the user-created virtual object locations and the real-world coordinates on an *a priori* static map of the service area available to the robot. The performance of each approach is evaluated and reported. We also present use-case scenarios to demonstrate potential applications of the AR-SRS for mobile service robots.**

I. Introduction

Mobile service robots are becoming increasingly popular in domestic environments to perform various household chores. The complexity and changing nature of domestic situations pose challenges in the operation of service robots for executing scheduled tasks. In such a dynamic environment, human communication can support robots to acquire and update spatial knowledge about locations of allocated tasks. More specifically, in some scenarios an operator may need to regularly interact with a robot to share spatial information for performing various tasks. For example, a user may want to direct a robotic vacuum cleaner according to the context and cleaning needs of various spaces [1], such as to frequently visit high-traffic areas that need additional cleaning or to avoid low-occupancy areas that do not require cleaning. Alternatively, building a semantic map of an environment can assist a user to operate a service robot by describing places from the human point-of-view. As in [2], user interactions can aid in the construction of semantic information for a service robot by labeling various rooms in an office environment. In such cases, an HRI system that can account for real-world environments while offering an easy-to-use and efficient mode of communication is necessary.

In prior research, numerous HRI modalities have been proposed to facilitate effective interaction with mobile robots for conveying to them location information for varied application scenarios. Specifically, some intuitive approaches have included laser pointers [3], [4], hand pointing gestures [5]–[7], and touchscreen interactions on a live video feed of the robot workspace [8]–[10], among others. In addition to providing natural modes of interaction, the laser pointers and pointing gestures enable users to interact with a robot from their own perspective view. However, these prior approaches require either smart environments with external sensors, such as RGB-D cameras [7] or on-board sensors of robots [3]–[6], to capture the user's intended gestures.

Alternatively, prior research [11] has suggested intuitive AR approach for sharing user intentions through dialogues in the form of spatial visual cues. AR offers mechanisms to integrate digital content into a real-world space for sharing spatial information in varied applications [12]. Not surprisingly, several AR-based approaches have been developed to mediate spatial information from human operators to mobile robots. In one conventional approach, an overhead camera captures the exocentric view of the robot's entire working environment [8]–[10], [13]. Next, by tracking unique markers affixed on the robot, the pose of the robot can be detected in its operating environment. Special markers are utilized to define or annotate location-specific information [14]. The pose of virtual objects superimposed on markers is used to perform designated operations such as motion control of robot or manipulation of objects [13], [15], [16]. In recent research, the overhead cameras and computer monitors have been replaced by a single see-through mobile AR device that acts both as the sensing and interaction device [15]–[17]. In yet other works, markerless AR technology is employed using Google Tango device for applications such as teaching virtual boundaries [18] and authoring tasks [19] to a mobile robot.

Even as the aforementioned techniques ensure a natural or intuitive communication with robots, a majority of them necessitate the use of structured settings [7]–[10] or specialized research-grade equipment [14], making their integration into existing robotic platforms or real-world environments challenging. For example, installing multiple overhead cameras to cover the entire operating environment of a mobile robot will be prohibitive when considering cost and privacy concerns. Moreover, overhead camera views are prone to problems arising from occlusion [19]. Similarly, for approaches relying on special markers, a key drawback

Work supported in part by the National Science Foundation under a DRK-12 grant DRL-1417769, an ITEST grant DRL-1614085, and an RET Site grant EEC-1542286, and a NY Space Grant Consortium grant 76156-10488.

[1]Mechatronics, Controls, and Robotics Laboratory (MCRL), Mechanical and Aerospace Engineering, NYU Tandon School of Engineering, Brooklyn, NY 11201, USA [sonia.mary, Armando.Granado, arajkumar, vkapila]@nyu.edu

is the practical difficulty of affixing a sufficient number of markers and creating a prepared environment for the robot operation [20]. Furthermore, the performance of marker-based approaches largely depends on the accuracy of marker tracking system. Thus, it is imperative that the markers be placed within a fixed distance from the tracking camera [21], which can limit its application to varied scenarios arising in human-robot shared workspaces. Prior approaches based on laser pointers or human gestures require computer vision techniques that rely on laser tracking systems or depth camera sensors, which further increases the overall cost of the robotic systems. Thus, in designing an HRI system, it is essential to consider some key factors such as its viability in real-world environments, portability to existing robotic platforms, cost burden, and scalability to various application scenarios while aiming for a user-friendly mode of interaction.

To address these challenges, in this paper, we propose a mobile AR solution that allows a human operator to interface with any ROS-enabled mobile robot and effectively communicate spatial information to it. We exploit the AR technology to intuitively specify and select any real-world surface location by creating AR virtual objects with semantic labels. The virtually marked locations are transformed to corresponding real-world locations in the coordinate frame of the mobile robot. To do so, we develop three different approaches that link the virtual object location and its intended real-world coordinates. These approaches support sharing of spatial locations to a co-located or remotely located mobile robot. With the use of smartphones as the *only* interaction device, the AR-SRS can be easily integrated to an existing mobile robot platform without any additional hardware cost. Furthermore, the interface can be tailored for a range of real-world mobile robot applications by customizing the AR application (see section III.C).

In this paper, we present the development, implementation, and testing of the AR-SRS for a ROS-enabled mobile robot. The rest of the paper is organized as follows. Section II outlines three approaches used in developing the AR-SRS and use-case scenarios. Section III discusses the experiments to examine the performance of each approach and analyzes the results. Finally, section IV concludes the paper and discusses future work.

II. Approach

An overview of the complete system is given in Fig. 1. The AR-SRS comprises of two main components: (*i*) a mobile AR application as the user interface for specifying spatial information and (*ii*) a ROS system (remote PC running ROS master) that performs necessary computations and storage of the spatial data. We developed the mobile AR application by adopting the state-of-the-art AR technology of Google's ARCore[2] that supports both markerless and marker-based AR technologies. With the AR application, a user can create virtual objects with semantic labels at any desired locations on the surface plane of operation of the mobile robot.

[2]https://developers.google.com/ar/reference/

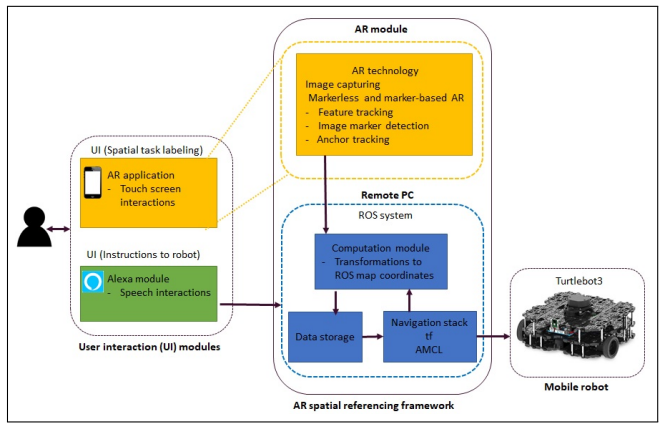
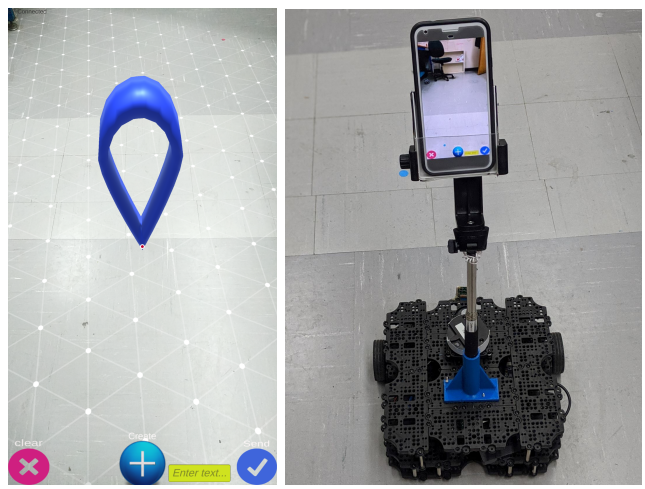


Fig. 1: Overview of the AR-SRS: User creates virtual objects with semantic labels at desired locations with the AR application. ROS master running on a PC receives the spatial data, performs computations, and stores results. User gives instructions using Alexa by calling the semantic labels for the robot to move to commanded locations. See demo video at http://engineering.nyu.edu/mechatronics/videos/arsrs.html.



(a) (b)

Fig. 2: (a) Screen-shot of AR application–center-button (+) creates virtual object; right-button (✓) sends spatial data to the PC; left-button (×) clears the virtual object and label; yellow color input-field used to enter label information and (b) AR device mounted on the robot and remotely operated for approach 2.

After being launched by the user, the AR application begins by scanning the surrounding environment and detects feature points and surface planes. Once a surface plane has been detected, the user is able to select any real-world location on the detected surface through the AR application. To do so, the user adjusts the placement of a red-dot target at the intended location and clicks the center button (+) on the AR application as shown in Fig. 2a. The pose of the real-world location is measured by performing a raycast against the detected surface plane. Specifically, a ray is projected from the cross-hair center on the phone screen onto the view space of the camera and the pose of the ray intersecting with the surface plane is returned. An AR anchor is instantiated

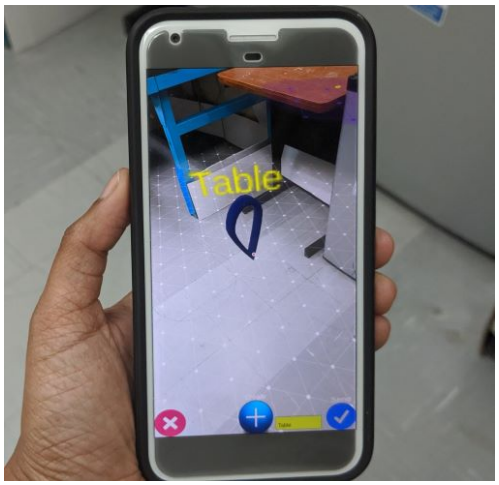


Fig. 3: Spatial referencing: User creates virtual object with a label on a desired location

on the corresponding location and a virtual object is created and attached to the anchor at that location. Next, the user can assign a label to the recently created virtual object by typing a text on the yellow color input-field (see Fig. 3).

Upon the press of the right-button (✓), the location and associated label (spatial data) of the virtual object are transmitted by the AR application using Bluetooth communication to the PC running the ROS master. The spatial data of locations in the AR environment is then mapped into corresponding locations of an *a priori* occupancy grid map (OGM) of the environment and stored in a database. The stored spatial data is retrieved later for instructing the robot to perform a specific task at the user-specified location. The following subsection describes the details of the AR-SRS framework and its integration with a ROS-enabled mobile robot. Moreover, the implementation of an Alexa speech module to assign tasks to the robot by enunciating one or more stored semantic labels is described.

A. AR spatial referencing system

The AR-SRS enables the user to mark and store information concerning any indoor or outdoor real-world spatial surface location by creating a virtual object at that location. The spatial data of a virtual object in the AR scene is transformed into the corresponding real-world location coordinate $M(X, Y)$ of a 2D static OGM. The location coordinates and associated label ('label' : $[X, Y]$) are stored in a database. A shared frame of reference is used to make correspondence between the AR environment and the OGM. We developed three different approaches to transform the virtual object's spatial data to the corresponding real-world coordinates of the OGM.

Approach 1: Image marker as the shared frame of reference This approach utilizes a 2D image marker (fiducial marker) at a known location on the map as a reference frame shared by both the OGM and AR-based simultaneous localization and mapping (SLAM), as shown in Fig. 4. The AR application creates an image anchor at the detected image marker location. This image anchor is tracked and used for

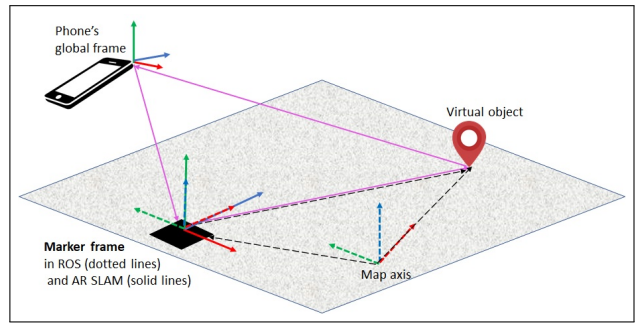


Fig. 4: Approach 1: Image marker as the shared frame of reference. The fixed marker frame links the virtual object location to its actual real-world coordinate on the map.

localizing the AR device (a smartphone) within the robot's map as the user moves with the phone on the mapped area. A spatial location is marked by the user creating a labeled virtual object as shown in Fig. 3. The location of the virtual object in the OGM is computed using the known pose of the image marker and the pose of the phone relative to the image marker. To do so, initially, the global pose of the newly created virtual object $({}^{\text{global}}P_{\text{object}})$ is converted into the marker's coordinate frame $(\mathcal{F}^{\text{marker}})$ and then into the robot map's coordinate frame $(\mathcal{F}^{\text{map}})$ by a set of transformations. This approach is adopted from our previous work with manipulator robots and detailed steps of transformations are similar to those available in [20]. The accuracy of the estimated virtual object poses in the OGM is primarily dependent on the accuracy of the AR system to track fiducial marker anchor and virtual objects.

Approach 2: Instantaneous pose of phone as the shared frame of reference In this approach, instantaneous pose of the phone is used to estimate the virtual object location on the map of the area (OGM). The phone is mounted on the robot with a fixed frame of reference in the robot's coordinate system in ROS (see Fig. 2b). The robot is localized on the map using adaptive Monte Carlo localization (AMCL), and the pose of the phone relative to the map is calculated based on the robot's pose. The transformation between map and phone frame is provided by the `tf` messages published by ROS. The phone frame, which is shared by both the robot and AR scene (see Fig. 5), is used to estimate the corresponding location of virtual objects in the OGM. The real-world location of the phone where the AR application is launched serves as the global coordinate frame of the phone in the AR scene. The AR application determines the pose of virtual objects based on this global frame of the phone. For computing the virtual object location relative to the robot's map, first the virtual object location in the global frame $({}^{\text{global}}P_{\text{object}})$ needs to be transformed into the phone's current frame $({}^{\text{pc}}P_{\text{object}})$. Next, the location of virtual object measured in the phone's coordinate frame in AR system $(\mathcal{F}^{\text{phone}_{(\text{AR})}})$ needs to be converted to phone's coordinate frame in the robot system $(\mathcal{F}^{\text{phone}_{(\text{robot})}})$. Finally, the absolute location of virtual object on the map $({}^{\text{map}}P_{\text{object}})$ is calculated by transforming the virtual object location in the phone's coordinate frame in the robot system to the map's

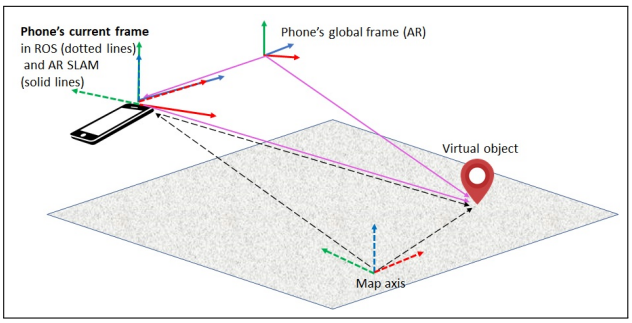


Fig. 5: Approach 2: Current pose of phone as the shared reference frame. The phone is mounted on the robot and its instantaneous coordinate frame links the virtual object location to its actual real-world coordinate on the map.

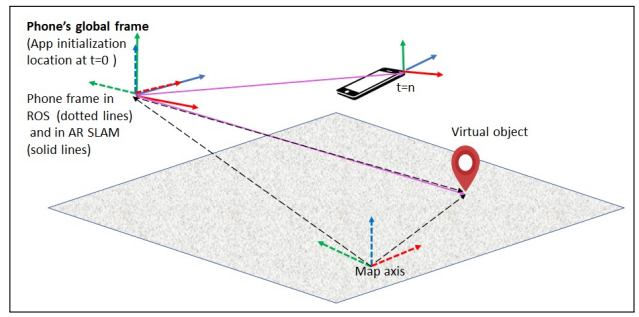


Fig. 6: Approach 3: Initial pose of phone as the shared frame of reference. With the phone mounted on the robot, when the AR application starts, the global (initial pose) frame of phone links the virtual object location to its actual real-world coordinate on the map.

coordinate frame. The detailed steps are given below and the transformations in the AR system are shown in pink color and in robot system as dotted lines in Fig. 5.

Step 1: Compute the virtual object location in the phone's current coordinate frame in the AR system

$$({}^{\mathrm{pc}}P_{\mathrm{object}})_{\mathrm{AR}} = \left({}^{\mathrm{global}}T_{\mathrm{pc}}\right)\left({}^{\mathrm{global}}P_{\mathrm{object}}\right). \quad (1)$$

Step 2: Transform the virtual object in the current left-handed coordinate system (LHS) of phone in the AR system to that of the right-handed coordinate system (RHS) of phone in the robot system.

$$Q = (R_{\mathrm{X}}(90^\circ))\,({}^{\mathrm{pc}}P_{\mathrm{object}})_{\mathrm{AR}}, \quad (2)$$

$$({}^{\mathrm{pc}}P_{\mathrm{object}})_{\mathrm{robot}} = EQ, \quad (3)$$

where R_{X} denotes the rotation matrix corresponding to a rotation about the X-axis and E is the elementary row operator to swap the coordinate axis for LHS to RHS coordinate transformation given by $E = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$.

Step 3: Compute the virtual object location in the robot's map frame

$${}^{\mathrm{map}}P_{\mathrm{object}} = ({}^{\mathrm{map}}T_{\mathrm{pc}})\,({}^{\mathrm{pc}}P_{\mathrm{object}})_{\mathrm{robot}}. \quad (4)$$

For this approach, the phone is to be always mounted on the robot to track its instantaneous location on a previously mapped area, hence it is suitable for a system that is to be operated remotely. The user can operate the robot from the remote PC (or a second phone) by keyboard control and the phone mounted on the robot is mirrored and seen (controlled) from the remote desktop PC (or the second phone). To mark a physical location, the user needs to move the robot close to that location so that the desired spot is visible on the phone screen.

Approach 3: Initial pose of phone as the shared frame of reference In this approach, the initial pose of the phone is utilized as the shared frame of reference between the AR and robot systems to estimate the corresponding coordinates of the virtual object on the map (OGM). As in Approach 2, the robot is localized on the mapped area, and the pose of the phone on the map is computed based on the robot's pose. The phone is initially (when the AR application starts) mounted on the robot with a fixed frame of reference in the robot's coordinate system. The global axis in the AR scene is the real-world location of the phone where the AR application starts. In other words, the location of the virtual object in global axis (${}^{\mathrm{global}}P_{\mathrm{object}}$) is same as that in the phone's initial coordinate frame (${}^{\mathrm{pi}}P_{\mathrm{object}}$), i.e., ${}^{\mathrm{global}}P_{\mathrm{object}} = {}^{\mathrm{pi}}P_{\mathrm{object}}$ (see Fig. 6). Once the AR application starts, the user can take off the mounted phone from the robot and walk around the area to mark any locations by creating virtual objects. The global pose of the virtual object is transformed into the map's coordinate frame through a set of transformations given below.

Step 1: Transform the global pose of virtual object in the AR system $({}^{\mathrm{pi}}P_{\mathrm{object}})_{\mathrm{AR}}$ to the initial pose of phone in the robot system $({}^{\mathrm{pi}}P_{\mathrm{object}})_{\mathrm{robot}}$ as in (2).

Step 2: Compute the virtual object location in the map's frame

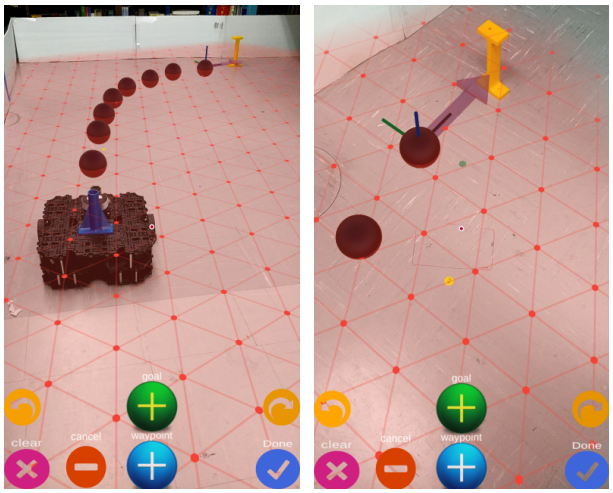
$${}^{\mathrm{map}}P_{\mathrm{object}} = ({}^{\mathrm{map}}T_{\mathrm{pi}})({}^{\mathrm{pi}}P_{\mathrm{object}})_{\mathrm{robot}}. \quad (5)$$

In this approach, after initially localizing the phone on the mapped area, phone is removed from the robot and the user can carry it around. Thus, the user gets a direct view of the scene while operating the AR application. Moreover, unlike in approach 2, the user doesn't require to move the robot with the mounted phone to create virtual objects.

B. ROS implementation

The mobile robot used for this work is a ROS-enabled TurtleBot3.[3] The workspace of the robot is mapped and a 2D OGM is created with gmapping algorithm using odometry and the on-board laser scan data [23]. The robot is localized using the AMCL approach, which uses a particle filter to track the pose of the robot against the OGM [24]. The remote ROS system receives the spatial semantic data of the virtual object using a serial node, performs necessary computations for transformations indicated previously, and stores the results as coordinates of the OGM. An Alexa voice module is integrated in the AR-SRS for instructing

[3] http://emanual.robotis.com/docs/en/platform/turtlebot3/overview/



(a) (b)

Fig. 7: Screen-shot of AR application: (a) robot navigates through a predefined virtual path and (b) the pose of robot at goal location is defined with a manipulable virtual object with direction arrows

the robot to perform a task at a stored spatial location by specifying the associated semantic label. The corresponding spatial information is retrieved and sent as a goal location on the map for executing the user-specified task at that location.

Alexa integration Amazon Alexa[4] voice service was utilized to send labeled spatial information to the Turtlebot3. With Amazon Web Services, a custom skill is developed to send instructions to the remote PC running the ROS master. The skill is activated by setting the invocation name as *AR Robot*. The user activates Alexa's function by saying "*Alexa, open AR Robot.*" After the skill is launched, the user gives instructions to the TurtleBot3. For example, the user says "*Go to Room 101.*" The words following the *Go to* are extracted as keywords and sent to the ROS master along with the specific skill word. A ROS node subscribes to the keywords sent by Alexa and retrieves the corresponding spatial information stored with the same keyword.

C. Possible Use-Cases

Predefined navigation path: In some domestic and warehouse situations, an autonomous mobile robot may be required to follow a predefined path to avoid any unexpected collisions as well as to provide safety and comfort to the people in the surrounding environment. Using the proposed AR-SRS, a virtual path can be quickly created in any dynamic environment. For example, construction sites continually change with project progress, and any newly introduced barriers in the robot path can be circumvented using this method. To construct a predefined navigation route, we have developed an AR application that creates a virtual path with via-points, as shown in Fig. 7a. A specialized manipulable virtual object marks the goal location that defines the orientation of the robot at the destination. Two yellow buttons (↺ and ↻) are provided in the AR application to modify the orientation of the goal object (see Fig. 7b).

[4]https://developer.amazon.com/en-US/alexa

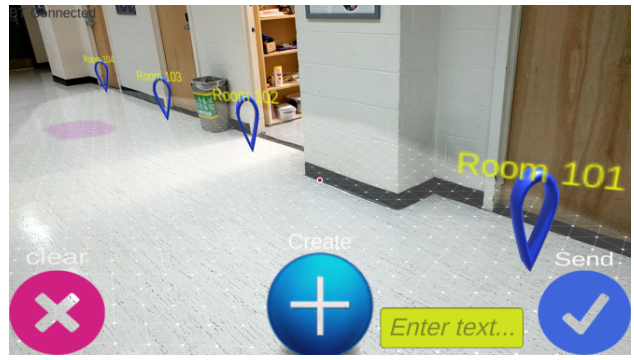


Fig. 8: Screen-shot of AR application: Room locations are marked by virtual objects with corresponding labels

Labeled goal locations: Users may find it convenient to communicate with a robot about a desired location by enunciating the location name. With the proposed AR-SRS, any location can be explicitly marked and labeled. For example, a delivery robot in an office environment can be instructed to go to a particular room by saying the room number. Any number of room locations and their names can be defined using the proposed approach, as shown in Fig. 8. Moreover, such an approach can be useful in home environments where the user can teach a service robot about various task locations by marking them with unique virtual object labels. The labeled location can be a safe zone for the robot that can be easily accessed through commands. Moreover, a prohibited area can be marked by creating boundaries with virtual objects to inform the robot to avoid that area.

III. System Evaluation and Discussion

A. Experiment setup

Experiments were conducted to evaluate the performance of the AR-SRS by creating and estimating spatial locations with each approach. Two random locations were selected and marked on the ground as reference points P_1 and P_2. The coordinates of $P_1 = (-0.3\text{m}, 0.3\text{m})$ and $P_2(0.9\text{m}, -0.1\text{m})$ were determined based on the ground truth location of the map (OGM) coordinate axis. The true location of the map origin is determined based on the odometry axis of the robot at the beginning of gmapping. We conducted 30 trials at each reference point by a single user with variations for each approach to evaluate the accuracy and repeatability of the system. In Approach 1 (A1) that relies on a marker, a 13.2cm $\times$ 13.2cm fiducial marker was affixed at a known location (at coordinate (0,0) for the experiment) in the map's coordinate frame. The user created labeled virtual objects at locations (P_1 and P_2), standing in different positions, with the desired location clearly visible and in close proximity of the phone, within one meter [22]. In Approach 2 (A2) that uses the shared instantaneous pose of the phone, the user created a virtual object at P_1 (and P_2) from different poses of the robot with the mounted phone. In Approach 3 (A3) that uses shared initial pose of the robot, the AR application was initialized by keeping the robot at a fixed pose. The user then walked around the mapped area with the phone to

TABLE I: Performance test results

	A1	A2	A3
P_1(X,Y) (m)	$(-0.298 \pm 0.010, 0.290 \pm 0.013)$	$(-0.306 \pm 0.007, 0.303 \pm 0.013)$	$(-0.311 \pm 0.003, 0.282 \pm 0.003)$
P_2(X,Y) (m)	$(0.900 \pm 0.011, -0.102 \pm 0.010)$	$(0.916 \pm 0.028, -0.081 \pm 0.014)$	$(0.890 \pm 0.011, -0.078 \pm 0.006)$
Accuracy (%)	2.54	3.62	3.92
Repeatability (CV %)	2.26	2.64	1.02

create virtual objects at P_1 (and P_2) from different standing positions. The experiment was repeated with different initial poses of the robot with the mounted phone.

B. Results

The performance of the AR-SRS is examined by evaluating the accuracy and repeatability of each approach. The accuracy of each approach is determined by utilizing the measurements of virtual points created by the user corresponding to the intended reference points P_1 and P_2. We normalize the user-specified virtual points corresponding to P_1 (and P_2) with the distance of the corresponding reference point P_1 (and P_2) from the map origin to report the accuracy of each approach (i.e., we transform all measurements to correspond to the unit displacement of reference points). The mean (μ) and standard deviation (σ) of measured values corresponding to P_1 and P_2 are provided in Table I. The accuracy test is performed in accordance with ISO 9283 standard [25]. The accuracy is reported as the percent deviation from the reference point values. It is seen that A1 yields slightly better accuracy with a $< 3\%$ deviation compared to A2 and A3 with a $< 4\%$ deviation. This indicates that the phone localization is slightly better with the marker-based approach compared to the other two approaches. These results indicate that the accuracy of the AR-SRS to specify a location is comparable to or better than that reported by some prior gesture pointing approaches [3], [6].

While mean and standard deviation are dependent on the actual quantities being measured, to holistically determine the repeatability of the three approaches we compute the percentage coefficient of variation: a unit-less quantity as given below. In Table I, the repeatability of each approach is expressed as the percentage coefficient of variation ($CV \triangleq \frac{\sigma_{\mathrm{N}}}{\mu_{\mathrm{N}}} \times 100\%$) of the normalized Euclidean distance computed for each measured point from the map origin (μ_{N} and σ_{N} denote the mean and standard deviation of normalized measurements corresponding to P_1 and P_2).

C. Discussion

There are different sources of errors affected by the estimation of virtual points at a desired real-world location, such as operator error (error in placing virtual object), tracking error of AR system (motion tracking and anchor tracking), and localization error of robot (induced by AMCL). In A1, the localization of the phone on the mapped area mainly depends on the accuracy of the AR device. However, in A2 and A3, the accuracy of the localization of the phone depends on the robot's localization accuracy. As indicated above, since the AMCL offers a limited pose accuracy it leads to the propagation of error in localizing the robot, as it moves away from its initial pose, and thus causing the build-up in the phone localization error. To reduce the effect of this cumulative error, we positioned and initialized the robot for every repetition of the experiment with a new robot pose for approaches A2 and A3. Alternative approaches for reducing the localization error resulting from AMCL can further improve the accuracy [26].

Fig. 9 provides the scatter plots showing the spread of estimated virtual locations defined by each approach at P_1 and P_2. It is observed that the virtual locations defined by A2 are scattered more compared to A1 and A3. Moreover, the relatively low values of CV reported by A1 and A3 indicate that these two approaches offer better repeatability compared to A2 (See Table I). For A1 and A3, users hold the phone and get a direct view of the physical space that helps them to select the locations with better precision.

Among the three approaches, A1 provides better accuracy but it has an additional requirement of affixing a marker at a known location on the mapped area. Moreover, spatial referencing operation under A1 does not require a robot system with a mounted phone. Though A2 offers slightly less performance, it supports remote operation that is useful for remote applications. A3 eliminates the requirement of marker calibration step as in A1 and offers a direct view of the environment, but requires the robot localization step. Any of the three approaches can be implemented for a situation by considering the above factors.

The differences in mode of operation and calibration requirements make the three SRS approaches suitable for different application contexts. For instance, A1 supports direct user interactions on the AR device and doesn't require a robot during spatial referencing that makes it suitable for off-line operations. For example, A1 is an excellent choice to prepare a semantic map of an environment (as shown in Fig. 8) before deploying a robot there. Alternatively, for A2, the device needs to be mounted on the robot to perform spatial referencing, which makes it useful only for teleoperation. Hence, this can be beneficial in scenarios where robots need real-time assistance in acquiring spatial data. For example, when a user wants a robot to retrieve a newly placed object from a room; the robot can be directed to that room and the object's location (assuming that the object lies on a surface plane) can be referred precisely by creating a virtual object over the real-object location. Similarly, A3 supports some characteristics of A1 (direct interaction on the device) and A2 (necessity of a robot for calibration); hence it is more suitable for providing just-in-time spatial information. For instance, A3 can be utilized for a vacuum cleaner robot to specify locations that require extra cleaning or to prioritize the cleaning areas just before setting a cleaning operation.

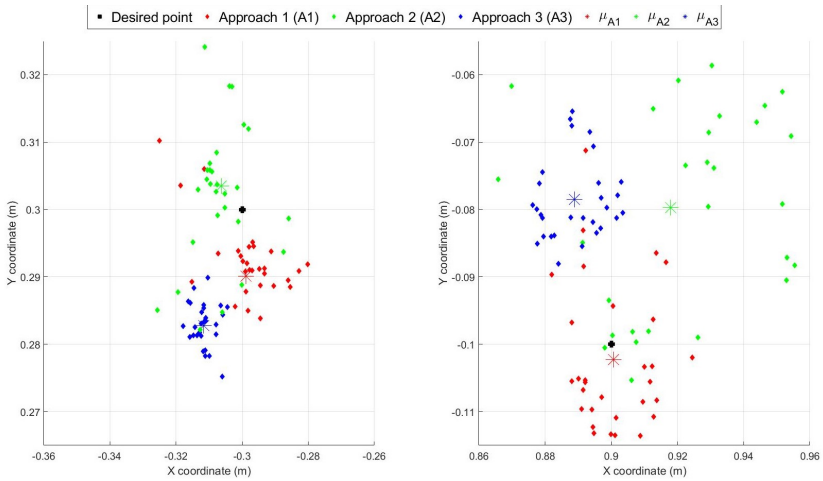


Fig. 9: Scatter plot of virtual point locations: At reference P_1 (left) and at reference P_2 (right)

IV. Conclusion and Future Work

We developed an interactive spatial referencing system to specify physical surface locations for allocating specific tasks for a mobile robot. With the proposed AR-SRS, users can flexibly define any location in the environment by creating and labeling a virtual object. We presented three approaches to transform the virtual object location to a corresponding location on a robot's map. The first method utilizes a known fiducial marker to localize the AR device, while the other two approaches are based on localization of the robot. The AR-SRS yields excellent accuracy with a $< 4\%$ deviation as well as high repeatability with a $< 3\%$ variation. We offered two use-case scenarios of the AR-SRS for a mobile robot. In future work, we will investigate potential applications of the AR-SRS by considering various requirements in a practical working environment of a service robot. We will also assess user perspectives by conducting qualitative user studies.

References

[1] H. Kim et al., "User-centered approach to path planning of cleaning robots: Analyzing user's cleaning behavior," *ACM/IEEE Int. Conf. on Human Robot Interaction* (*HRI*), pp. 373–380, 2007.

[2] C. Zhao, W.Mei, and W. Pan, "Building a grid-semantic map for the navigation of service robots through human–robot interaction," *Digital Communications and Networks*, 1(4), pp. 253–266, 2015.

[3] C.C. Kemp et al., "A point-and-click interface for the real world: Laser designation of objects for mobile manipulation," *ACM/IEEE Int. Conf. on Human Robot Interaction* (*HRI*), pp. 241–248, 2008.

[4] K. Ishii et al., " Designing laser gesture interface for robot control," *IFIP Conf. on Human-Computer Interaction* (*INTERACT*), pp. 479–492, 2009.

[5] S. Abidi, M. Williams, and B. Johnston, "Human pointing as a robot directive," *ACM/IEEE Int. Conf. on Human Robot Interaction* (*HRI*), pp. 67–68, 2013.

[6] M. Tolgyessy et al., "Foundations of visual linear human–robot interaction via pointing gesture navigation," *Int. J. of Social Robotics*, 9(4), pp. 509–523, 2017.

[7] R. Rasch et al., "Tidy up my room: Multi-agent cooperation for service tasks in smart environments," *J. of Ambient Intelligence and Smart Environments*, 11(3), pp. 261–275, 2019.

[8] D. Sakamoto et al., "Sketch and run: A stroke-based interface for home robots," *SIGCHI Conf. on Human Factors in Computing Systems* (*CHI*), pp. 197–200, 2009.

[9] J.A. Frank, Y. Sahasrabudhe, and V. Kapila, "An augmented reality approach for reliable autonomous path navigation of mobile robots," *Indian Control Conf.* (*ICC*), pp. 437–443, 2015.

[10] J.A. Frank and V. Kapila, "Path bending: Interactive human-robot interfaces with collision-free correction of user-drawn paths," *Int. Conf. on Intelligent User Interfaces* (*IUI*), pp. 186–190, 2015.

[11] P. Milgram et al., "Applications of augmented reality for human-robot communication," *Int. Conf. on Intelligent Robots and Systems* (*IROS*), vol. 3, pp. 1467–1472, 1993.

[12] S.A. Green et al., "Human-robot collaboration: A literature review and augmented reality approach in design," *Int. J. of Advanced Robotic Systems*, 5(1), pp. 1–18, 2008.

[13] H. Sunao et al., "TouchMe: An augmented reality based remote robot manipulation," *Int. Conf. on Artificial Reality and Telexistence* (*ICAT*), 2011.

[14] D. Sprute et al., "A framework for interactive teaching of virtual borders to mobile robots," *IEEE Int. Symp. Robot and Human Interactive Communication* (*RO-MAN*), pp. 1175–1181, 2007.

[15] S. Kasahara et al., "exTouch: Spatially-aware embodied manipulation of actuated objects mediated by augmented reality," *ACM Int. Conf. on Tangible, Embedded, and Embodied Interaction* (*TEI*), pp. 223–228, 2013.

[16] J.A. Frank, S.P. Krishnamoorthy, and V. Kapila, "Toward mobile mixed-reality interaction with multi-robot systems," *IEEE Robotics and Automation Letter*, vol. 2, pp. 1901–1908, 2017.

[17] L. Chen et al., "PinpointFly: An egocentric position-pointing drone interface using mobile AR," *SIGGRAPH Asia Emerging Technologies*, pp. 34–35, 2019.

[18] D. Sprute, K. Tönnies, and M. König, "Virtual borders: Accurate definition of a mobile robot's workspace using augmented reality," *Int. Conf. on Intelligent Robots and Systems* (*IROS*), pp. 8574–8581, 2018.

[19] Y. Cao et al., "V.Ra: An in-situ visual authoring system for robot-IoT task planning with augmented reality," *Designing Interactive Systems Conference* (*DIS*), pp. 1059–1070, 2019.

[20] S.M. Chacko and V. Kapila, "An augmented reality interface for human-robot interaction in unconstrained environments," *Int. Conf. on Intelligent Robots and Systems* (*IROS*), pp. 3222–3228, 2019.

[21] S. Zollmann et al., "Flyar: Augmented reality supported micro aerial vehicle navigation," *IEEE Trans. on Visualization and Computer Graphics* (*TVCG*), 20(4), pp. 560–568, 2014.

[22] S.M. Chacko and V. Kapila, "Augmented reality as a medium for human-robot collaborative tasks," *IEEE Int. Symp. Robot and Human Interactive Communication* (*RO-MAN*), 2019.

[23] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with Rao-Blackwellized particle filter," *IEEE Tran. on Robotics*, 23(1), pp. 34–46, 2007.

[24] D. Fox, "Adapting the sample size in particle filters through KLD-sampling," *The Int. J. of Robotics Research*, 22(12), pp. 985–1003, 2003.

[25] ISO 9283, *Manipulating Industrial Robots. Performance Criteria and Related Test Methods*, Int. Organization for Standardization, 1998.

[26] J. Roewekaemper et al., "On the position accuracy of mobile robot localization based on particle filters combined with scan matching," *Int. Conf. on Intelligent Robots and Systems* (*IROS*), pp. 3158–3164, 2012.