RF-Compass: Robot Object Manipulation Using RFIDs

Jue Wang   Fadel Adib   Ross Knepper   Dina Katabi   Daniela Rus

Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology

{jue_w, fadel, rak, dk, rus}@mit.edu

ABSTRACT

Modern robots have to interact with their environment, search for objects, and move them around. Yet, for a robot to pick up an object, it needs to identify the object's orientation and locate it to within centimeter-scale accuracy. Existing systems that provide such information are either very expensive (e.g., the VICON motion capture system valued at hundreds of thousands of dollars) and/or suffer from occlusion and narrow field of view (e.g., computer vision approaches).

This paper presents RF-Compass, an RFID-based system for robot navigation and object manipulation. RFIDs are low-cost and work in non-line-of-sight scenarios, allowing them to address the limitations of existing solutions. Given an RFID-tagged object, RF-Compass accurately navigates a robot equipped with RFIDs toward the object. Further, it locates the center of the object to within a few centimeters and identifies its orientation so that the robot may pick it up. RF-Compass's key innovation is an iterative algorithm formulated as a convex optimization problem. The algorithm uses the RFID signals to partition the space and keeps refining the partitions based on the robot's consecutive moves. We have implemented RF-Compass using USRP software radios and evaluated it with commercial RFIDs and a KUKA youBot robot. For the task of furniture assembly, RF-Compass can locate furniture parts to a median of 1.28 cm, and identify their orientation to a median of 3.3 degrees.

Categories and Subject Descriptors C.2 [**Computer Systems Organization**]: Computer-Communications Networks

General Terms Algorithms, Design, Performance, Theory

Keywords Robot Mobile Manipulation, RFID, RF Localization, Optimization

1. INTRODUCTION

Today's robots routinely replace human labor in assembly tasks [34, 43]. However, the substantial cost and complexity involved restrict their use to large factories with high-volume pro-

duction lines. There, the robots are typically bolted to a bench at a known location, and one type of parts is repeatedly presented to them, one item at a time with a fixed positioning and orientation. Such highly specialized assembly robots are cost-effective only in the presence of economies of scale. This leaves small to medium size factories and manufacturers unable to leverage robot operations to automate their production. Hence, it has been of a great interest in both the robotics academic community and industry to enable robots to perform more flexible and complex tasks, such as assembling products [26], fetching and delivering items [7], and collaborating with other robots [31]. For example, it is desirable to have a robot automatically fetch the various components of IKEA furniture and assemble them. In contrast to robots on large-volume assembly lines, such a robot could be deployed to assemble different types of furniture in a store.

Such more capable, adaptive robots must interact with their environment in the presence of uncertainty. They need to search for a desired object, pick it up, and move it around. These tasks require locating the object to centimeter-scale accuracy and identifying its orientation to within a few degrees so that the robot may grasp it. Currently, these functions rely on technologies that are expensive and/or error-prone. For instance, recent research demonstrating robots assembling an IKEA table [31] relies on the infrared VICON motion capture system, which costs hundreds of thousands of dollars [47]. Other solutions based on computer vision [52] and depth-imaging [38] require learning visual models of the desired objects and can fail in distinguishing visually similar objects or partially occluded objects [52]. Further, all existing technologies, whether infrared or vision-based, fail in the absence of a line-of-sight to the object, e.g., furniture part covered by other parts.

In comparison to infrared and vision-based approaches, emerging RF-based localization systems [50, 41, 6] offer an appealing alternative, with the advantages of being cost-effective and applicable to occluded and non-line-of-sight objects. Further, the wireless signal can encode the object ID, enabling the robot to easily identify a desired object without complex machine learning algorithms based on object shape and color. There are two main challenges, however, in using RF localization in robot navigation and object manipulation tasks. First, although many robots today already have WiFi connectivity, it is impractical to also place WiFi nodes on every object the robot needs to interact with. Second, the accuracy achieved by current RF localization techniques is not sufficient for object manipulation and grasping. For example, fetching the correct IKEA table leg without knowing its placement in the environment a priori requires an accuracy of position and orientation on the order of a few centimeters and degrees, respectively.

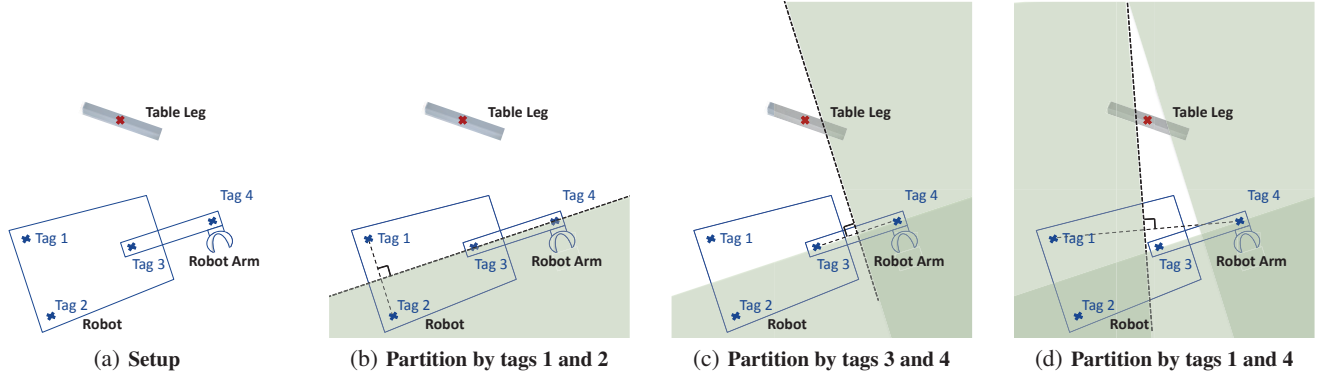| (a) **Setup** | (b) **Partition by tags 1 and 2** | (c) **Partition by tags 3 and 4** | (d) **Partition by tags 1 and 4** |

Figure 1—Illustration of RF-Compass's partitioning algorithm: By comparing the distances from the red RFID to a pair of blue RFIDs on the robot, we can partition the space into two regions, one of which contains the furniture part. Overlaying these partitions narrows down the candidate region. Further, the motion of the robot naturally leads to different ways of partitioning the space, which when combined iteratively refine the candidate region as the robot navigates.

This paper introduces RF-Compass, an RFID-based system that navigates and adjusts the actions of the robot to centimeter accuracy. In RF-Compass, ultra-low power, low cost RFIDs are placed on both the robot and the objects it needs to work on. In contrast to WiFi nodes, RFIDs (5–10 cents each) can be easily embedded into physical objects, as articulated in the vision of the Internet of Things [3]. In fact, many industries are already rapidly deploying RFIDs on their products as a replacement to barcodes, which do not work in non-line-of-sight settings [19, 18, 46].

RF-Compass's key innovation is a new algorithm capable of leveraging the robot's consecutive moves to iteratively refine its location and orientation estimates. Our algorithm uses pairwise comparison of RFID signals to partition the space into regions according to their distances to the desired object. Suppose we have four tags on the robot and one tag on the object as in Fig. 1(a). By comparing whether the object is closer to tag 1 than to tag 2, we can partition the space into two regions; one of them contains the desired object, as shown in Fig. 1(b). We can repeat this process to check if tag 3 or tag 4 is closer to the RFID on the object. This will generate a second partition, which when laid on top of the first partition, adds specificity to the solution, as in Fig. 1(c) and then 1(d). Mathematically, each relation between a pair of tags on the robot and the tag on the object translates into a linear constraint. We can incorporate these constraints into an optimization problem that bounds the region where the object is located.

Comparing its distances to a few RFIDs on the robot may not bound the object into a centimeter-scale region. The strength of the algorithm comes from that every move of the robot gives us new partitions, which we can lay over the old partitions to obtain a finer region for the desired object. As the robot gets close, it can maneuver its arm above the object to calibrate the size of the region to the desired accuracy. Furthermore, by having more than one RFID on the object itself, we can apply the same partitioning algorithm to each of the tags on the object to discover the orientation of the object and pinpoint its center.

An important feature of this algorithm is robustness to potential errors in estimating the RFIDs' pairwise distances. Specifically, the optimization returns the region that is consistent with the maximum number of pairwise constraints and ignores inconsistent constraints. Correct pairwise distances are consistent with each other because they all reflect the actual layout of the RFIDs. In contrast, errors are typically random and inconsistent with each other, and hence can be overcome by optimizing for consistency.

Summary of Results: We built a prototype of RF-Compass using software radios and commercial UHF RFIDs. We integrated the system with ROS, the Robot Operating System [40] which has support for robot navigation and path planning. We used our prototype to navigate a KUKA youBot robot [32] and direct its actions in grasping an IKEA table leg in both line-of-sight and occluded scenarios. We compared RF-Compass with the VICON motion capture system used in past work on robot furniture assembly [31]. VICON is a highly accurate system that provides millimeter positioning information [47], but costs hundreds of thousands of dollars and works only in instrumented rooms where the whole ceiling is fitted with infrared cameras and the robot and objects are equipped with intrusive infrared-reflective markers.

Our experiments lead to the following findings:

- Across experiments, the length of the path traversed by an RF-Compass-guided robot to reach the object (an IKEA table leg) is on average only 6% longer than the optimal path, and at most 16% longer than the optimal path.
- Using two RFIDs on the object, RF-Compass's median accuracy of estimating the center position and orientation is 2.76 cm and $5.77°$. Increasing the number of RFIDs on the object to four, RF-Compass can pinpoint its center and orientation to a median of 1.28 cm and $3.3°$. These results enable commercial robots with a variety of hand gripper sizes to grasp an object, while leaving a sufficient margin for error, which we validated via trace-driven simulations.
- The VICON infrared system cannot locate the furniture part in occluded settings (e.g., the part is under a chair). Hence, it fails to navigate the robot in these cases. In contrast, RF-Compass's performance is consistent in both line-of-sight and non-line-of-sight.

Contributions: This paper makes the following contributions:

- RF-Compass is the first RF localization system that supports robot object manipulation and grasping. It enables robots to leverage RF signals to locate an object to within a few centimeters and identify its orientation to within a few degrees so that the

robot may grasp it. As such, RF-Compass bridges robotics with RF localization and solves a real problem for the robotics community, who needs localization solutions that are highly accurate, cost-effective, and capable of dealing with occlusion.

- RF-Compass introduces a new RF localization algorithm formulated as a space partitioning optimization problem. The new algorithm has three features: 1) it can leverage the robot's consecutive moves to refine the accuracy of its estimates; 2) it is robust to potential errors in pairwise comparisons of RFID distances; and 3) it estimates the object's orientation in addition to its location.

- Finally, RF-Compass is implemented and evaluated in a testbed of RFIDs and robots, demonstrating the practicality of the design.

2. RELATED WORK

Enabling robots to navigate to objects in their environment and grasp them is a fundamental problem in robotics, commonly referred to as mobile manipulation. In this context, RF-Compass facilitates two operations: object detection and centimeter-scale localization. Past works that address these problems may be divided into two categories. The first category requires significant instrumentation of the environment. In particular, the VICON motion capture system [47], widely used in robotics [31, 35], requires the deployment of many infrared cameras (costing hundreds of thousands of dollars) on the ceiling of the room of operation. It also requires adding intrusive infrared-reflective markers to the objects as well as the robot. The output of this system, however, is highly accurate to the millimeter scale.

The second category of work requires no major modification of the environment and mainly relies on optical cameras [42, 13] or depth imaging (e.g., Kinect, LIDAR) [38, 12]. These systems must solve the object detection problem using machine vision algorithms. Thus, they typically require prior training to recognize specific objects [42, 38]. When successful at detecting an object, they can locate it with an error of a few centimeters to tens of centimeters, depending on the amount of clutter and the system capability [28, 15]. The challenge in these systems is that variations in background, lighting, scene structure, and object orientation exacerbate an already difficult computer vision problem, and can cause these systems to miss the desired object [21, 15].

In comparison to the above systems, RF-Compass is relatively cheap and does not require heavy instrumentation of the environment. It naturally solves the object detection problem by leveraging the unique EPC [16] IDs of the deployed RFIDs, and hence eliminates the need for complex computer vision software. Also, since RF signals propagate in non-line-of-sight, it can address the occlusion problem which is common to all of the above approaches, including VICON.

Prior papers have used RF signals, both WiFi and RFID, for localization and robot navigation [6, 22, 29, 30, 5, 45, 14, 41, 50, 27] but not for grasping objects. The typical accuracy of these systems is a fraction of a meter to a few meters. Although they are adequate for fuzzier localization tasks like detecting which room a robot is in, they cannot support high-precision tasks like object grasping and manipulation.

Several past proposals can achieve higher localization accuracy at the cost of covering the whole floor with a dense, surveyed grid of RFID tags that are spaced at a few to tens of centimeters [39, 23, 44, 11, 49]. Among these proposals, the closest to our work is PinIt [49], which employs dynamic time warping (DTW) to compare RF signals in order to map a tagged object to its nearest neighbor among the deployed reference RFIDs. RF-Compass builds on
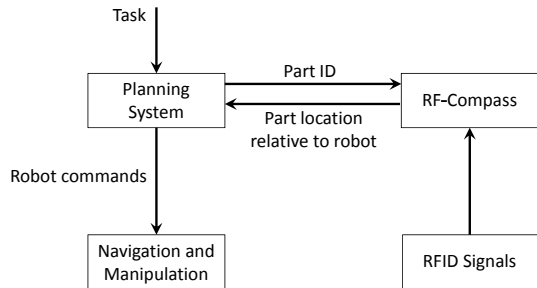

Figure 2—System Architecture: A task planning system directs the robot to perform various actions in order to accomplish a higher-level goal, such as fetching all parts of a piece of furniture and assembling it. RF-Compass interfaces with the task planning system to estimate the location of individual parts needed by the system.

this past work but differs from all proposals in this category along four major axes: First, RF-Compass does not require a deployment of reference RFIDs in the environment. Second, RF-Compass introduces a novel RF localization algorithm formulated as a space partitioning optimization that leverages the robot's consecutive moves to iteratively refine the accuracy of the optimization. Third, RF-Compass has built-in robustness to errors in identifying the nearest neighbor based on comparing RF signals. Finally, RF-Compass is significantly more accurate than past work and can localize an object to within a couple of centimeters and identify its orientation to within a few degrees, hence enabling robotic object manipulation and grasping.

3. OVERVIEW

RF-Compass is an RFID-based solution for enabling robots to navigate towards an object of interest and manipulate it. Although we present the system in the context of the furniture assembly task, RF-Compass's technique applies to a variety of robotic applications, including other assembly tasks, fetching and delivery, and robot collaboration.

3.1 Scope

Robot mobile manipulation tasks, e.g., delivery and assembly, involve multiple modules that work together to complete the task. These modules include: 1) a task *planner* that has a blueprint of a sequence of actions to be performed by the robot in order to complete the task; 2) a *navigator* that controls and commands the robot's wheels and hand to move towards a specified destination; and 3) an *object detection and positioning* system that locates the desired object and computes its orientation so that the robot may navigate towards it and grasp it. Each of these modules on its own spans a research area. RF-Compass focuses on the last module, i.e., it provides a centimeter-scale solution for object detection and localization. It is easy to integrate with standard task planners and navigators. For our empirical evaluation, we have leveraged the planner and navigator functions available in ROS, the Robot Operating System [40].

RF-Compass's design assumes that one can attach UHF RFIDs to the body and hand of the robot, as well as the objects of interest. We believe this assumption is reasonable since UHF RFIDs are cheap and naturally designed to be adhered to products. We also assume that the RFIDs on the robot and the furniture parts are within the reading range of the RFID reader. Today's UHF RFIDs have an operating range of around 10 m [53, 48]. Thus, the assumption is
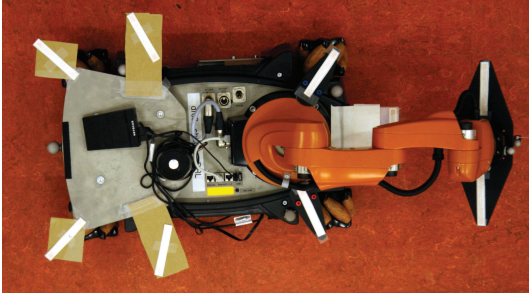
Figure 3—The manipulator robot: In our experiments, the KUKA youBot has 9 UHF RFIDs deployed on its hand and base.

satisfied in common cases where the delivery and assembly task is performed in a confined space.

3.2 System Architecture

We provide a brief overview of how RF-Compass fits within a standard robot mobile manipulation framework, illustrating the components in our system and their interaction. A diagram of the system architecture is shown in Fig. 2. The input to the system is a high-level task, such as "fetch all parts and tools needed for table assembly". A task planner receives this input and constructs a sequence of actions to be performed by the robot, such as "fetch table leg 3". It then queries RF-Compass about a particular object "table_leg_3".

Given the part identifier ("table_leg_3") from the task planner, RF-Compass looks up in its database the RFID tags on the part and their positions within the part itself. Then, RF-Compass queries both the RFIDs deployed on the robot and the ones on the part. Based on the RFID signals collected, RF-Compass estimates the center position and the orientation of the part with respect to the robot.

RF-Compass returns to the task planner its estimate of the part's relative location to the robot as a tuple (direction ϕ, distance D, orientation θ).[1] Based on this relative location, the planner calculates the robot motion needed. The planner issues instructions to the navigation and manipulation component, which commands the robot to move accordingly.

The task planner, the navigator, and RF-Compass reside on one computer, and communicate with the robot over a WiFi link, a setup commonly used in modern robots [40].

4. RF-COMPASS

In RF-Compass, we deploy a set of RFIDs on the manipulator robot. For example, for the youBot which has a base of 0.58 m by 0.38 m and an arm of length 0.65 m, we deploy a total of 9 RFIDs, as shown in Fig. 3. We also attach several (e.g., 1–4) RFIDs to each of the manipulated objects.

RF-Compass uses an iterative algorithm, where each iteration involves the following steps:

1 *Measuring relative proximity:* RF-Compass queries the RFIDs on the robot and the manipulated object. It uses the RFID signals to learn, for each tag on the object, whether it is closer to tag i or tag j on the robot.

---

[1]RF-Compass's estimate of the orientation θ is only meaningful after the part is within a short distance from the robot, which is needed for the final steps of positioning the robot hand for grasping.
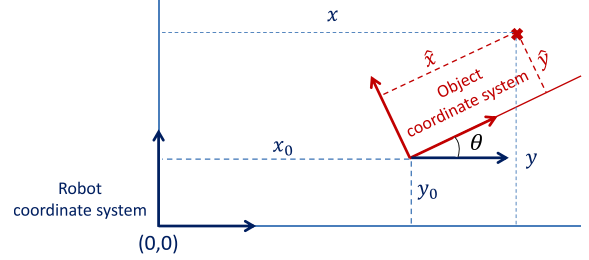

Figure 4—Robot's and object's coordinate systems: The robot and the object of interest each have their own coordinate systems. The origin of the object coordinate system is represented as (x_0, y_0) in the robot's coordinate system, and its orientation is θ. The position of an RFID tag can be expressed in the robot's coordinate system as (x, y), or in the object's coordinate system as $(\hat{x}, \hat{y})$.

2 *Localization by partitioning:* RF-Compass feeds this pairwise relative proximity information as constraints to a convex optimization problem that partitions the space into regions according to the relative proximity information.

3 *Refinement:* The partitions are overlaid over previously learned partitions to keep refining the resolution. The algorithm stops when RF-Compass can identify the object's center and orientation with sufficient precision. Otherwise, the robot moves towards the partition that contains the object, and the algorithm proceeds to the next iteration.

Below we describe these steps in detail.

4.1 Measuring Relative Proximity

RF-Compass bootstraps its algorithm with questions of the form: given a pair of RFID tags on the robot, which of the two tags is closer to a particular RFID tag on the object? The answers to these questions provide a list of inequalities of the form $d_{T_k^P, T_i^R} \leq d_{T_k^P, T_j^R}$, where T_i^R and T_j^R are two RFIDs on the robot, T_k^P is an RFID on the object, and $d_{T_k^P, T_i^R}$ is the distance between tag T_k^P and tag T_i^R. RF-Compass then uses these inequalities as constraints in an iterative space-partitioning algorithm that estimates the location of the desired object to within a couple of centimeters and its orientation to within a few degrees.

The literature has multiple proposals that RF-Compass may leverage for obtaining such bootstrapping information. They estimate the relative distance between pairs of RF sources by comparing the RSSI [36], angle of arrival (AoA) [37, 4], or the multipath profile of their signals [49]. In RF-Compass, we use the approach in [49], which has been shown to be more robust to multipath effects and non-line-of-sight propagation. To do so, we use an antenna array on the RFID reader to receive the RFID signals. Such an antenna array can be implemented cheaply using a single sliding antenna by employing the technique of synthetic aperture radar (SAR) [17, 49]. We then estimate a measure of similarity between the signals of two RFIDs using dynamic time warping (DTW). Higher DTW-similarity between two signals is associated with a shorter distance.

We note that RF-Compass can deal with potential errors in the above relative proximity information. Specifically, RF-Compass obtains a large number of pairwise comparisons in the form of $d_{T_k^P, T_i^R} \leq d_{T_k^P, T_j^R}$, creating a highly over-constrained partitioning problem. The valid constraints are coherent, i.e., they agree on a common region for the tag on the object, while the outliers tend to be random and inconsistent with each other. In §4.3.A, we explain

Term	Definition
(x_0, y_0)	the origin of object coordinate system (center of object) expressed in robot coordinate system
θ	the orientation of the object coordinate system in the robot coordinate system
$T_1^R, ..., T_N^R$	the N RFIDs deployed on the robot
$(x_1^R, y_1^R), ..., (x_N^R, y_N^R)$	positions of $T_1^R, ..., T_N^R$ expressed in the robot coordinate system
T^P	the RFID at the object center, whose position is (x_0, y_0) in the robot coordinate system
$T_1^P, ..., T_K^P$	the K RFIDs deployed on object
$(\hat{x}_1^P, \hat{y}_1^P), ..., (\hat{x}_K^P, \hat{y}_K^P)$	positions of $T_1^P, ..., T_K^P$ expressed in the object coordinate system
$(x_1^P, y_1^P), ..., (x_K^P, y_K^P)$	positions of $T_1^P, ..., T_K^P$ expressed in the robot coordinate system
$d_{T_k^P, T_i^R}$	Euclidean distance between RFIDs T_k^P and T_i^R
ε	a confidence threshold used in outlier rejection
$\mathbf{z}$	the vector to be estimated in basic optimization with a single RFID on the object $\mathbf{z} = [x_0, y_0]^T$
$\mathbf{z}'$	the vector to be estimated in optimization using multiple RFIDs on object $\mathbf{z}' = [x_0, y_0, \cos\theta, \sin\theta]^T$
(x_0^{new}, y_0^{new})	the origin of object coordinate system expressed in the new robot coordinate system
θ^{new}	the orientation of the object coordinate system in the new robot coordinate system
(x_m, y_m)	the origin of the new robot coordinate system expressed in the old robot coordinate system
θ_m	the orientation of the new robot coordinate system expressed in the old robot coordinate system

Table 1—Terms used in the description of RF-Compass.

how RF-Compass performs outlier rejection by seeking the solution that agrees with the largest number of constraints.

4.2 Localization by Space Partitioning

The relative proximity information provides constraints on the distances between the tags on the robot and those on the object. We feed these constraints to a convex optimization solver that locates the object via iterative space partitioning. Before we describe the partitioning algorithm, we discuss the coordinate systems used in expressing locations. For reference, all terms used in our algorithm are defined in Table 1.

4.2.A Coordinate Systems

RF-Compass locates objects from the perspective of the robot – i.e., it expresses locations in a Cartesian coordinate system whose origin is at the center of the robot, and the three axes are oriented according to the robot's front-back, left-right, and up-down lines. For clarity, we will omit the up-down coordinates and describe the system in two dimensions.

RF-Compass exploits the fact that it knows the placement of the RFIDs deployed on the robot, e.g. two RFIDs are placed on the left and right of the robot's hand. It likewise knows the placement of the tags deployed on the object with respect to the object itself. For example, IKEA may deploy two RFIDs at the two ends of the table leg during manufacturing, and provide this information as input to RF-Compass. Given this information, RF-Compass estimates the position and the orientation of the object in the robot's coordinate system. Note that the placement of the RFIDs on the object is typically provided with respect to the object itself, such as one tag at each end of a table leg. Thus, we need a transform to convert from the object's coordinate system to the robot's coordinate system.

Fig. 4 illustrates the relation between these two coordinate systems. The origin of the object's coordinate system is at position (x_0, y_0) in the robot's coordinate system, i.e., as perceived by the robot. The object's coordinate system is rotated by θ relative to the robot's coordinate system.

We denote the coordinates of an RFID in the robot's coordinate system as (x, y), and its coordinates expressed in the object's system as $(\hat{x}, \hat{y})$. Given (x_0, y_0) and θ, one can convert between the two coordinate systems as follows:

$$\begin{aligned} x = \ & x_0 + \hat{x}\cos\theta - \hat{y}\sin\theta \\ y = \ & y_0 + \hat{x}\sin\theta + \hat{y}\cos\theta \end{aligned} \tag{1}$$

In RF-Compass, we define the origin of the object's coordinate system to be the center of the object. For any tag $T_k^P, k \in \{1, ..., K\}$ deployed on the object, we know its position in the object's coordinate system $(\hat{x}_k^P, \hat{y}_k^P)$, but do not know its coordinates from the perspective of the robot (x_k^P, y_k^P). For any tag $T_i^R, i \in \{1, ..., N\}$ deployed on the robot, we know its position (x_i, y_i) in the robot's coordinate system. Based on this set of information and the pairwise relative proximity information discussed in §4.1, RF-Compass estimates the center of the object (x_0, y_0) and its orientation θ in the robot's coordinate system as explained below.

4.2.B Convex Optimization for a Single Robot Position and a Single Tag on the Object

Although our algorithm is iterative, this section considers a single iteration with a fixed robot location. We focus our discussion on the basic case where one RFID T^P is deployed at the center of the object and N RFIDs $T_1^R, ..., T_N^R$ are deployed at $(x_1, y_1), ..., (x_N, y_N)$ on the robot.

Recall that d_{T^P, T_i^R} is the Euclidean distance between object tag T^P and robot tag T_i^R. RF-Compass uses the relative proximity information from §4.1:

$$d_{T^P, T_i^R} \leq d_{T^P, T_j^R}$$

$$(x_0 - x_i)^2 + (y_0 - y_i)^2 \leq (x_0 - x_j)^2 + (y_0 - y_j)^2$$

$$[2(x_j - x_i) \quad 2(y_j - y_i)] \cdot \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \leq x_j^2 + y_j^2 - x_i^2 - y_i^2 \tag{2}$$

The right side of the inequality and the row vector on its left side are both known, since we know the placement of the tags on the robot. The unknown variables are $[x_0, y_0]^T$, which RF-Compass aims to estimate.

Given N tags on the robot, we have $\frac{N(N-1)}{2}$ equations similar to Eq. 2, which form a matrix inequality

$$\mathbf{A}\mathbf{z} \leq \mathbf{b}, \tag{3}$$

where $\mathbf{z} = [x_0, y_0]^T$. $\mathbf{A}$ is a matrix of size $\frac{N(N-1)}{2} \times 2$, each row of which is $[2(x_j - x_i) \quad 2(y_j - y_i)]$ for different pairs of i and j. $\mathbf{b}$ is a
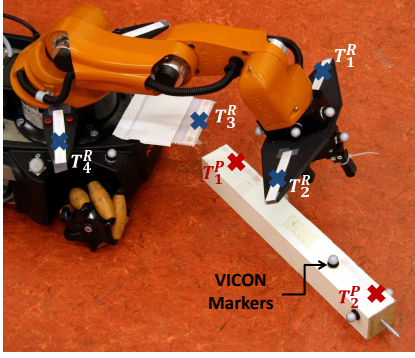
Figure 5—Robot arm and object: The figure shows the robot arm and hand equipped with RFIDs marked in blue, and the IKEA table leg with RFIDs marked in red. The silver balls on the robot and the table leg are the VICON infrared-reflective markers (VICON is used as a baseline).

column vector, where each element is $x_j^2 + y_j^2 - x_i^2 - y_i^2$ for different pairs of i and j.

Eq. 3 defines a linear feasibility problem, which can be rewritten in the following convex optimization formulation:[2]

$$\begin{aligned} \underset{\mathbf{z}}{\text{minimize}} \quad & 0 \\ \text{subject to} \quad & \mathbf{A}\mathbf{z} \le \mathbf{b} \end{aligned} \tag{4}$$

Many efficient methods exist to solve for $\mathbf{z}$ given $\mathbf{A}$ and $\mathbf{b}$. In our implementation, we use a solver based on the interior-point method [20]. The solver uses logarithmic barrier functions to approximate linear inequality constraints and returns the center of the feasible region. In this basic formulation, RF-Compass can use the returned $\mathbf{z} = [x_0, y_0]^T$ to navigate the robot in the direction of $\phi = \arctan \frac{y_0}{x_0}$. At the end of each iteration, RF-Compass has an estimate of the distance to the object $D = \sqrt{x_0^2 + y_0^2}$. The robot conservatively moves by $\frac{D}{2}$ in each iteration, until the object is within its reach, which can be decided by comparing the average distance between the robot tags themselves and their distances to the object tag.

4.2.C Generalizing to Multiple Tags on the Object

Aside from navigating towards the object, the robot also needs to know the exact center position and the orientation of the object so as to perform manipulation tasks such as grasping a particular part of the object. In order to estimate the orientation and pinpoint the center, we deploy $K > 1$ RFIDs $T_1^P, ..., T_K^P$ at $(\hat{x}_1^P, \hat{y}_1^P), ..., (\hat{x}_K^P, \hat{y}_K^P)$ on the object. In our evaluation, we experiment with $K = 2$ to 4 tags in estimating the object orientation.

Since the long arm of the robot extends outside its base as Fig. 5 shows, once the object is within 0.6 m of the robot's body center, the tags installed on the robot's hand and the ones on its base will start to appear at different sides of the object. Fig. 6(a) shows the topology of the same RFIDs as in Fig. 5 when viewed from above, where T_1^R and T_2^R are the two tags next to the robot's hand and T_3^R and T_4^R are two tags at the front of its base. The IKEA table leg has two RFIDs deployed at its two ends.

Taking the object tag T_1^P alone, we can determine that it is in the white partition in Fig. 6(a), using the basic optimization in *4.2.B*.

Similarly, taking T_2^P, we can identify that it is in the white partition in Fig. 6(b). Overlaying the two figures together produces Fig. 6(c). Since we know the dimensions of the IKEA table leg and that T_1^P and T_2^P are at the two ends of it, we can leverage the relationship between the two tags to jointly solve an optimization problem. In Fig. 6(c), imposing the constraints based on the distance between T_1^P and T_2^P allows us to better confine the orientation and center position of the object. Next we explain how to incorporate such constraints into our optimization problem.

Using the coordinate transform in Eq. 1, we have

$$\begin{aligned} x_k^P = \quad & x_0 + \hat{x}_k^P \cos \theta - \hat{y}_k^P \sin \theta \\ y_k^P = \quad & y_0 + \hat{x}_k^P \sin \theta + \hat{y}_k^P \cos \theta \end{aligned} \tag{5}$$

For each tag on the object T_k^P, RF-Compass evaluates its relative proximity to each pair of robot tags T_i^R and T_j^R:

$$d_{T_k^P, T_i^R} \le d_{T_k^P, T_j^R}$$

$$(x_k^P - x_i)^2 + (y_k^P - y_i)^2 \le (x_k^P - x_j)^2 + (y_k^P - y_j)^2$$

Replacing x_k^P and y_k^P using Eq. 5 leads to:

$$\begin{bmatrix} 2(x_j - x_i) \\ 2(y_j - y_i) \\ 2(x_j - x_i)\hat{x}_k^P + 2(y_j - y_i)\hat{y}_k^P \\ 2(y_j - y_i)\hat{x}_k^P - 2(x_j - x_i)\hat{y}_k^P \end{bmatrix}^T \begin{bmatrix} x_0 \\ y_0 \\ \cos \theta \\ \sin \theta \end{bmatrix} \le x_j^2 + y_j^2 - x_i^2 - y_i^2 \tag{6}$$

Note that we know all elements in the row vector on the leftmost in Eq. 6, since the placement of the robot tags is known and the placement of the tags on the object with respect to the object's center is known. Similarly, we also know the right side of the inequality. The unknown variables are now x_0, y_0 and θ, the center position and the orientation of the object in the robot's coordinate system.

Given K tags on the object and N on the robot, we have $\frac{KN(N-1)}{2}$ equations as Eq. 6, which form a matrix inequality:

$$\mathbf{A'}\mathbf{z'} \le \mathbf{b}, \tag{7}$$

where $\mathbf{z'} = [x_0, y_0, \cos \theta, \sin \theta]^T$. $\mathbf{A'}$ is a matrix of size $\frac{KN(N-1)}{2} \times 4$, each row of which is the row vector in Eq. 6 for different k, i, and j. $\mathbf{b}$ is the same column vector as in the basic formulation before.

$\mathbf{A'}$ and $\mathbf{b}$ are known to RF-Compass. If we consider $\cos \theta$ and $\sin \theta$ as two variables, then Eq. 3 defines a feasibility problem linear in $\mathbf{z'}$. In order to leverage the relationship between various tags on the object, we introduce another constraint $\cos^2 \theta + \sin^2 \theta \le 1$,[3] and the formulation becomes:

$$\begin{aligned} \underset{\mathbf{z'}}{\text{minimize}} \quad & 0 \\ \text{subject to} \quad & \mathbf{A'}\mathbf{z'} \le \mathbf{b} \\ & \|z'(3)\|^2 + \|z'(4)\|^2 \le 1 \end{aligned} \tag{8}$$

Eq. 8 is still a convex optimization problem and can be solved using the same interior-point method as before. In fact, when the robot is fairly far away from the tag, Eq. 8 reduces to the basic formulation in Eq. 4. This is because all the tags on the object will lie in one partition defined by the robot tags. As a result, the interior-point method will estimate their positions to be the same, and hence the orientation of the object θ cannot be decided. In other words, the $\cos \theta$ and $\sin \theta$ estimated in $\mathbf{z'}$ via Eq. 8 are not meaningful when the object is far away. Once the object is within a certain distance to the robot, different tags on the object will lie in different regions. In this case, the joint optimization based on their relationship will lead to a solution that best fits the position and orientation of the object.

---

[2]It is standard to formulate a feasibility problem as an optimization by specifying a constant objective function [8].

[3]Note that we cannot impose the constraint as $\cos^2 \theta + \sin^2 \theta = 1$, because such an equality is non-convex.

(a) **Partition for T_1^P**   (b) **Partition for T_2^P**   (c) **Constraining the center and orientation**
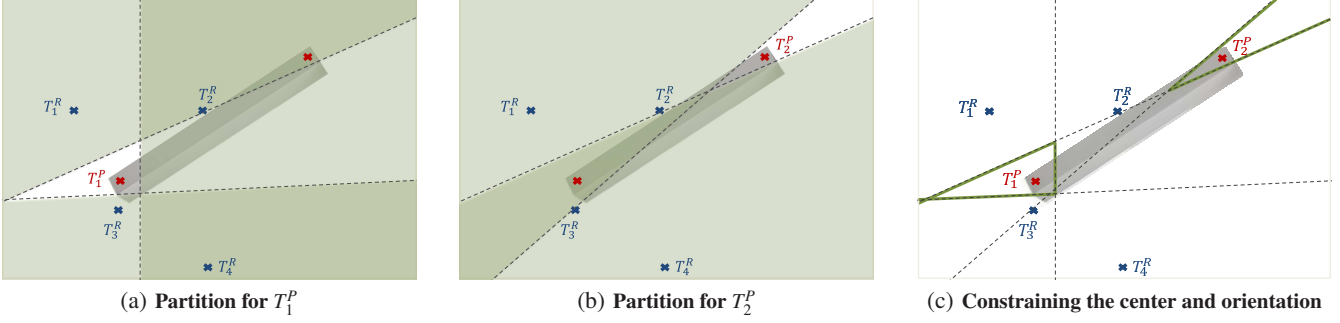
Figure 6—Utilizing multiple tags on object: (a) and (b) show the partitions obtained by individually locating T_1^P and T_2^P using the basic optimization in Eq. 4. Leveraging the knowledge about the relationship between their positions, RF-Compass formulates a joint convex optimization problem, pinpointing the orientation and center position of the object.

4.3 Iterative Refining and Outlier Rejection

RF-Compass leverages the motion of the robot to iteratively refine the resolution of partitioning. Specifically, each motion of the robot (e.g., moving its base or adjusting its hand) is an iteration. Each location of the robot will result in a different set of partitions. By overlaying these partitions, we can obtain a larger number of smaller regions, i.e., a higher resolution of the space.

Translating this idea into an iterative algorithm, we formulate a larger convex optimization problem by integrating the constraints derived from several consecutive moves of the robot. Note, however, that as the robot moves, its own coordinate system changes. Since RF-Compass knows the motion of the robot, we can transform the different coordinate systems in different iterations to a unified one. Yet, due to minor wheel slips of the robot, odometry error accumulates as the robot moves more and more, which means the coordinate transforms will be less and less accurate. Hence, RF-Compass only combines the partitioning constraints created in three consecutive motions of the robot and across at most 2 m.

Without loss of generality we describe how to formulate a larger convex optimization problem by integrating over two iterations, i.e., two consecutive locations of the robot. Let us still use Eq. 6 to represent the linear constraints created at the first location. Now let us consider the linear constraints at the second location:

$$
\begin{bmatrix} 2(x_j - x_i) \\ 2(y_j - y_i) \\ 2(x_j - x_i)\hat{x}_k^P + 2(y_j - y_i)\hat{y}_k^P \\ 2(y_j - y_i)\hat{x}_k^P - 2(x_j - x_i)\hat{y}_k^P \end{bmatrix}^T \begin{bmatrix} x_0^{new} \\ y_0^{new} \\ \cos\theta^{new} \\ \sin\theta^{new} \end{bmatrix} \leq x_j^2 + y_j^2 - x_i^2 - y_i^2
$$

Since the robot has moved, the position and the orientation of the object with respect to the robot has changed: $\mathbf{z}'^{new} = [x_0^{new}, y_0^{new}, \cos\theta^{new}, \sin\theta^{new}]^T$ is different from $\mathbf{z}' = [x_0, y_0, \cos\theta, \sin\theta]^T$. However, the placement of the RFIDs on the robot with respect to the robot itself does not change; nor does the placement of the RFIDs on the object with respect to the center of the object. As a result, the right side of the inequality and the row vector on the leftmost are known, and can be calculated similarly as at the first location: $\mathbf{A}'^{new}\mathbf{z}'^{new} \leq \mathbf{b}^{new}$.[4]

We use x_m, y_m and θ_m to denote the origin and orientation of the coordinate system at the second robot location in the first coordinate system, i.e., the one defined in the first iteration. RF-Compass knows x_m, y_m and θ_m because we know the exact movement the robot performed. We can derive the following equation using the

---

[4]Note that $\mathbf{A}'^{new}$, $\mathbf{b}^{new}$ are different from $\mathbf{A}'$, $\mathbf{b}$ because the relative proximity information changes as the robot moves.

coordinate transform in Eq. 1:

$$
\mathbf{z}' = \mathbf{U} \cdot \mathbf{z}'^{new} + \mathbf{V}, \tag{9}
$$

where

$$
\mathbf{U} = \begin{bmatrix} \cos\theta_m & -\sin\theta_m & 0 & 0 \\ \sin\theta_m & \cos\theta_m & 0 & 0 \\ 0 & 0 & \cos\theta_m & -\sin\theta_m \\ 0 & 0 & \sin\theta_m & \cos\theta_m \end{bmatrix} \tag{10}
$$

and

$$
\mathbf{V} = [x_m, y_m, 0, 0]^T \tag{11}
$$

Since we have $\mathbf{z}'^{new} = \mathbf{U}^{-1} \cdot (\mathbf{z}' - \mathbf{V})$, we can rewrite $\mathbf{A}'^{new}\mathbf{z}'^{new} \leq \mathbf{b}^{new}$ as

$$
\mathbf{A}'^{new}\mathbf{U}^{-1} \cdot (\mathbf{z}' - \mathbf{V}) \leq \mathbf{b}^{new} \tag{12}
$$

$\mathbf{A}'^{new}, \mathbf{U}, \mathbf{V}$ and $\mathbf{b}^{new}$ are known, hence by now we have transformed the partitions created in the second iteration into linear constraints in the first iteration's coordinate system, with the same variable $\mathbf{z}'$. Thus, we can combine these linear constraints with the ones in the first iteration and solve them jointly. The resulting optimization has the same form as Eq. 7 except with more rows, and can be solved similarly.

Feasibility problems in the form of Eq. 12 are solved using linear programming and have a complexity of $O(rn^{2.5})$, where n is the number of linear constraints and r is the rank of the matrix $\mathbf{A}'^{new}\mathbf{U}^{-1}$ [51]. In our implementation, we use the CVX [20] MATLAB package to solve the convex optimization problem. While it is possible to reduce the running time by optimizing the code, the MATLAB code on a 64-bit laptop with Intel Core i5 processor takes on average 0.6 seconds, which is insignificant compared to the mechanical process of moving the robot.

4.3.A Outlier Rejection

The RF propagation environment can result in outliers in RF-Compass's relative proximity measurements, i.e., the optimization takes the constraint $d_{T_k^P, T_i^R} \leq d_{T_k^P, T_j^R}$ while tag T_k^P is in fact closer to tag T_j^R than to tag T_i^R. RF-Compass's outlier rejection works as below. On a high level, since the iterative algorithm is formulated as an over-constrained system with a large number of relative proximity constraints, the optimization problem will turn out to be infeasible due to the outliers. However, the majority of the relative proximity constraints are correct and will coherently converge. Thus, we can leverage this to reject the inconsistent outliers by seeking a solution that agrees with most of the constraints.
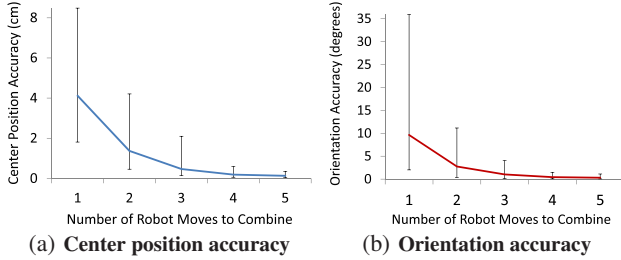
9

(a) **Center position accuracy**   (b) **Orientation accuracy**

Figure 7—Iterative refining by combining multiple moves of robot: This figure shows the median, 10th, and 90th percentile accuracy of RF-Compass's joint optimization, as we vary the number of robot moves combined. The median accuracy is around 4 cm when using a single location of the robot to partition the space. The accuracy quickly improves to within a fraction of a centimeter as the algorithm overlays the partitions obtained from consecutive moves of the robot.

In particular, since outliers are more likely to occur when the pair of distances being compared have a small difference, RF-Compass sets a predefined threshold ε and decides that tag T_k^P is closer to tag T_i^R than to T_j^R only if $m_{T_k^P, T_i^R} \leq (1 - \varepsilon) \cdot m_{T_k^P, T_j^R}$, where $m_{T_k^P, T_i^R}$ is the DTW distances measured between tags T_k^P and T_i^R [49]. The larger ε is, the more constraints we are ruling out from the optimization, and the more confident RF-Compass is about the remaining constraints. RF-Compass initializes the confidence threshold $\varepsilon = 0$, and increases it by 0.05 each time if the optimization problem is infeasible, until finding a solution. In our experiments, with 9 RFIDs on the robot and 2–4 RFIDs on the object, typically increasing ε to 0.05 or 0.1 is sufficient to solve the optimization and reject the outliers.

Using the above techniques to refine the estimation and reject outliers, RF-Compass directs the robot step by step to reach the object of interest and grasp it. RF-Compass decides that the robot's hand is well-positioned for grasping the object, when the proximity metric between the tags installed on the left and right of the hand and their counterparts on the object is below a threshold. At this point, RF-Compass's iterative partitioning algorithm terminates and lets the robot grasp the object.

4.4 Simulation Results

To understand the resolution that can be achieved using our optimization formulation of the partitioning algorithm, we perform a simulation with six RFIDs on the robot including the two next to its hand as in Fig. 5, and two RFIDs on the object. We focus on the accuracy of estimating the center position and the orientation of the object in the final steps of the algorithm. Hence, we assume the object is already within reach, e.g., 25 cm of the robot's hand, and check whether the algorithm can pinpoint the object's location to within a few centimeters and its orientation to within a few degrees, so that the robot may grasp it. In each trial, we randomly generate a location and orientation for the object. We derive the relative proximity constraints based on the positions of the RFIDs on the robot and the RFIDs on the object.

Impact of combining partitions across consecutive moves of the robot: We investigate the impact of iterative refinement on the performance of RF-Compass. We plot in Fig. 7 the center position and orientation accuracy of the iterative partitioning algorithm when it utilizes different numbers of consecutive robot moves to form the joint optimization. Fig. 7 shows the median accuracy as well as the
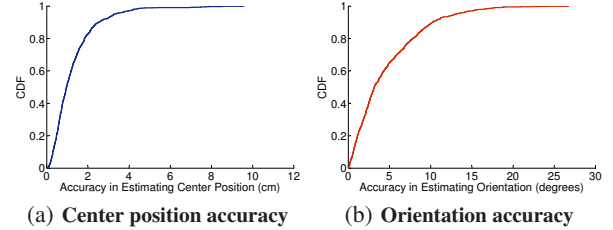

(a) **Center position accuracy**   (b) **Orientation accuracy**

Figure 8—Simulation results for the partitioning algorithm with 10% errors in the relative proximity information: Despite that 10% of the distance constraints are flipped randomly, the partitioning algorithm still achieves a centimeter scale location accuracy and can identify the orientation to within a few degrees.

10th and 90th percentiles (error bars) for 1000 runs. As we can see, the error of the center position estimate rapidly decreases as the number of combined robot moves increases. The median position accuracy is around 4 cm when using a single location of the robot, and quickly improves to within a fraction of a centimeter as we overlay the partitions obtained from multiple moves of the robot. This behavior is expected because every time we add a new location of the robot (i.e., a new iteration), the granularity of partitioning becomes finer. Similarly, the orientation accuracy also quickly improves as we combine partitions obtained from consecutive moves of the robot.

We note however that the simulation assumes no noise in odometry measurements. In practice, odometry errors accumulate as the robot moves. In order to limit this effect on the performance of RF-Compass's iterative algorithm, we only combine up to three moves of the robot.

Robustness to potential errors in relative proximity constraints: We investigate RF-Compass's outlier rejection. We flip 10% of the relative proximity constraints so that they are wrong. We then run the partitioning algorithm identifying the region that is consistent with the maximum number of pairwise constraints and plot in Fig. 8 the CDFs of the error in estimating the center and the orientation of the object. The CDFs are taken over 1000 runs.

Fig. 8(a) and 8(b) show that the median error in center position is 0.96 cm, and its 90th percentile is 2.5 cm. The median error in orientation estimation is only $3.1°$ and its 90th percentile is $10°$. These results verify that the partitioning algorithm is both accurate, and robust to potential errors in the relative proximity information. This robustness is due to that we are dealing with an over-constrained optimization. Errors tend to make the optimization infeasible, whereas all of the correct constraints have a consistent solution. Hence, eliminating the constraints that make the optimization infeasible tends to eliminate the errors.

5. IMPLEMENTATION

We built a prototype of RF-Compass using USRP software radios [25] and the Alien Squiggle RFIDs [2]. We used the prototype to direct the actions of a KUKA youBot [32] in reaching and grasping furniture parts, as detailed below.

Robot and Manipulated Objects: We use a KUKA youBot robot, which has four separately-controlled wheels and is capable of moving with arbitrary linear and angular velocity up to approximately 80 cm/sec and 1 rad/sec. The robot has a manipulator arm, which consists a serial chain of five links 65.5 cm in total length and five joints. The combination of the arm and the wheels is capable of performing essentially arbitrary motions in all dimensions.

(a) VERT900 6-inch Omni-directional Antenna

0.8 cm

(b) Alien Squiggle General Purpose UHF RFID

Figure 9—Antenna and commercial UHF RFID used in experiments: (a) We move the VERT900 6-inch vertical antenna in a straight line to emulate a linear antenna array. (b) the Alien Squiggle General Purpose UHF RFID Tags. The sizes of (a) and (b) are not proportional to the real sizes of the devices.

The arm can lift a payload of 0.5 kg, which is sufficient to assemble many products from IKEA. The youBot hand's maximum aperture (finger tip-to-tip distance) is 7.5 cm. In these experiments, the robot's goal is to grasp table legs measuring 5 cm wide, leaving a 2.5 cm margin of error tolerance.[5] The furniture parts are the legs of the IKEA Lack table [24]. The dimensions of the table legs are 40 cm×5 cm×5 cm.

RFIDs: We deploy the Alien Squiggle RFIDs [2] on both the robot and the furniture part. The Alien Squiggle Inlay shown in Fig. 9(b) is a commercial off-the-shelf passive UHF RFID widely used in supply chains, logistics and asset tracking operations. Each tag costs 5–10 cents. 9 RFIDs are placed on the robot and 1–4 RFIDs are placed on the table leg. Note that each RFID has a unique 96-bit EPC ID [16], and as a result we can easily distinguish different objects in a pile based on the unique IDs of the RFIDs attached to them.

RFID Reader and Antenna Array: We adopt a USRP implementation of an EPC Gen-2 RFID reader developed in [9] to query the RFIDs. We use RFX900 daughterboards and Cushcraft 900 MHz RFID reader antennas [33], and run all experiments at a carrier frequency of 920 MHz.

As mentioned in §4.1, we extract relative proximity information using the approach in [49]. Similarly to the work in that paper, we implement an antenna array on the RFID reader by using a single sliding antenna and post-processing the received signal with the SAR algorithm [17]. Specifically, one option for building an antenna array is to use multiple USRPs, synchronize them via an external clock, and have each of them act as one antenna in the array. This approach, however, will unnecessarily limit the array to a handful of antenna elements. Alternatively, and since we are working with robots, we can leverage motion to build an array with many antenna elements using a technique known as SAR (Synthetic Aperture Radar) [17, 49, 1]. To implement SAR, we mount a USRP RFID reader equipped with an omni-directional VERT900 antenna (shown in Fig. 9(a)) on a second robot, which moves with a constant speed in a straight line. As it moves, the antenna acts as if it were a different antenna situated at that point in space. Thus, this moving receive antenna emulates in time an array of antennas spaced along a line. One can then process the signal received by the moving antenna at different points in time as if it were the output of a spatial array of antennas, and apply standard antenna array equations (see [49] for more details on SAR).

The reader repeatedly queries the tags and uses the moving antenna to collect the response signals. We use these signals to extract

[5]In comparison, Baxter [42], like most commercial robots designed for manipulating objects, has more joints (seven), longer arms (104 cm), larger hands (>15 cm) and higher payload (2.3 kg) than the youBot, and thus can pick up a wider variety of objects.
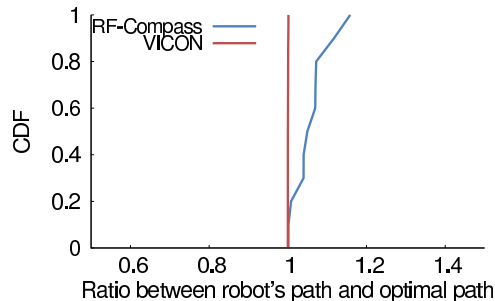

Figure 10—Ratio between the length of path traversed by the robot and the length of optimal path in line-of-sight: When directed by RF-Compass, the path traversed by the robot is on average only 6% longer than the optimal path, and at most 16% longer.

pairwise relative proximity information as explained in §4.1, which is fed to the partitioning algorithm.

Navigation and Manipulation: ROS [40] provides software stacks for navigation and manipulation. The navigation stack delivers real-time obstacle avoidance and the object manipulation stack provides control of a manipulator arm for picking up and moving objects.

Baseline: In [31], the VICON motion capture system is used to demonstrate letting a youBot fetch and deliver the same type of IKEA table legs. Thus, we use VICON as a baseline. We place both the robot and the table legs inside the VICON room and vary their initial distance in a range of 2–6 meters. The choice of the range is mainly dictated by the size of our VICON deployment where the maximum area reliably covered by the infrared cameras is 6×5 m^2.[6] VICON requires placing infrared-reflective markers on both the robot and the object of interest. Thus, we place 10 such markers on the robot and 4 markers on the table leg.

6. EVALUATION

RF-Compass is applicable to many mobile manipulation tasks, such as tool delivery, collecting wastes, collaborating with other robots, etc. Here, we evaluate it in a task related to furniture assembly. Specifically, the IKEA furniture assembly application in [31] involves robot collaboration where the delivery robots retrieve various furniture parts and hand them off to the assembly robots for installation. In our evaluation, we focus on the job of the delivery robot.

We study RF-Compass's performance in three phases:

- **Navigation:** In this phase, RF-Compass should detect the needed part and navigate the delivery robot towards it. By the end of this phase, the object is within the reach of the robot's hand. RF-Compass can determine that the furniture part is within reach when the RFIDs next to the robot's hand are closer to the RFIDs on the part than to the RFIDs at the end of the robot's base, as described in §4.1. For this phase, we evaluate the length of the navigation path traversed by the robot.
- **Localization:** The second phase is preparation for manipulation. For this phase, we evaluate whether RF-Compass can accurately identify the center of the part and its orientation to enable the robot to grasp it.
- **Object Manipulation:** The third phase is grasping. We evaluate the impact of RF-Compass's accuracy on the number of trials the robot needs before it can successfully grasp the part.

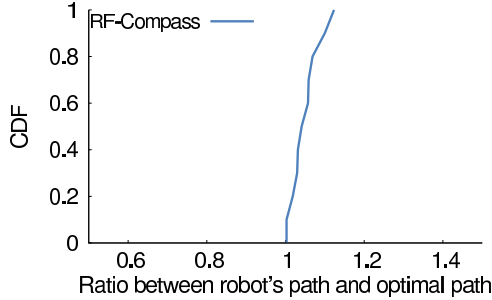[6]This deployment of VICON costs a few hundred thousand dollars.

Figure 11—Ratio between the length of path taken by the robot and the length of the optimal path in non-line-of-sight: When directed by RF-Compass, the path traversed by the robot is on average 5% longer than the optimal path, and at most 12% longer.

6.1 Efficiency of RF-Compass in Robot Navigation

We conduct 10 experiments in line-of-sight and another 10 experiments in non-line-of-sight, comparing using RF-Compass and VICON to navigate the robot towards the part. In each experiment, the robot and the furniture part are placed at different random initial locations and their initial distance is in a range of 2–6 meters. We compare the paths traversed by the robot using RF-Compass versus using VICON, for the same sets of initial locations. RF-Compass uses 2 RFIDs on the furniture part and 9 RFIDs on the robot, while the VICON system uses 4 infrared-reflective markers on the furniture part and 10 such markers on the robot.

Line-of-Sight Evaluation: In these experiment, we keep the emulated antenna array, the furniture part, and the robot all in direct line-of-sight of each other. We also ensure that both the robot and the part are within the reliable detection region of the VICON motion capture system and there is no occlusion of the VICON markers on the robot or the part.

Fig. 10 shows the CDF of the ratio between the length of the path traversed by the robot to that of the optimal path, for both RF-Compass and VICON. The optimal path has a length equal to the Euclidean distance between the robot's initial position and its destination. In this case, the VICON system can locate both the furniture part and the robot to millimeter accuracy, since there is no occlusion or clutter of the infrared-reflective makers. Hence, it always finds the optimal path. The path traversed by the robot as directed by RF-Compass is on average 6% longer than the optimal and at most 16% longer than the optimal path.

Non-Line-of-Sight Evaluation: In these experiments, we put up a 1.5 m×2.2 m solid wooden board supported by metal structures to block the line-of-sight for RF-Compass's emulated antenna array. We also create occlusion of the furniture part by placing it under an office chair. We further use desks, chairs, and other objects to introduce obstructions. In each experiment, we vary the placement of the obstructions.

Fig. 11 shows the CDF of the ratio between the length of the path traversed by the robot to that of the optimal path. Since the object is occluded from the sight of the infrared cameras, the VICON system cannot locate it and hence fails to provide an estimate of where the robot should move. This behavior is common in existing localization systems for object manipulation, including vision-based systems as well as infrared systems like VICON, LIDAR, and Kinect. In contrast, RFID signals can penetrate obstacles and reach the RFID reader even when the object is occluded. Thus, as in the line-of-sight experiments, with non-line-sight and occlusions, the path in RF-Compass is close to optimal.
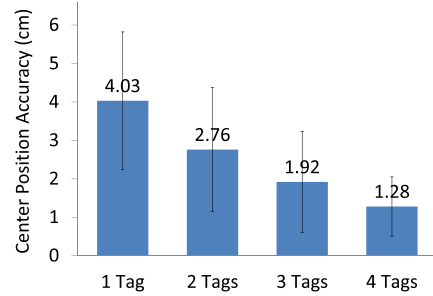

Figure 12—RF-Compass's accuracy in estimating center position: With 1 tag on the part, the error in RF-Compass's estimation has a median of around 4 cm. Leveraging 4 tags on the part, RF-Compass's median error reduces to 1.28 cm.
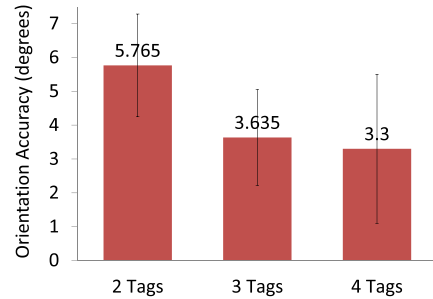

Figure 13—RF-Compass's accuracy in estimating object orientation: With two tags on the part, RF-Compass's median accuracy of the orientation estimate is around 6°, and further decreases to around 3° when 4 tags are used.

6.2 Accuracy of RF-Compass in Estimating Center Position and Orientation

Next, we evaluate RF-Compass's performance in mobile manipulation. At this stage, the object is already within the reach of the robot, e.g., 0.25 m from its hand. Thus, we focus on the accuracy RF-Compass can achieve in identifying the center of the part and its orientation. We experiment with 1–4 tags deployed on the IKEA table leg.

Center Position Accuracy: Fig. 12 shows the distance between RF-Compass's estimate of the center of the table leg and the ground truth captured by the VICON system.[7] With one tag at the center of the table leg, RF-Compass's median error in estimating the position is 4.03 cm, with a standard deviation of 1.8 cm. The accuracy improves to 2.76 cm when two tags are placed at the two ends of table leg, and further improves to 1.28 cm when four evenly spaced tags on the part are used in the partitioning algorithm.

Orientation Accuracy: In order to determine the orientation of an object, at least two RFIDs need to be placed on the object. Fig. 13 shows the difference between RF-Compass's estimate of the table leg's orientation and the ground truth. RF-Compass has a median orientation error of around 6° when two tags on the table leg are used. This error further decreases to around 3° when all four tags are used in the optimization.

---

[7]When the robot arm is above the part, sometimes it prevents the VICON cameras from seeing the infrared-reflective markers. To make sure our ground truth measurement of the part is accurate, after RF-Compass's iterative algorithm terminates, we move the robot away without moving the part, and let VICON record its position.

6.3 Number of Attempts in Grasping Furniture Part

Finally, we investigate the impact of RF-Compass's accuracy on the robot's ability to grasp objects. Specifically, we compute the number of attempts before the robot succeeds in grasping the leg of the IKEA table. Given the location and orientation of the object, the planner commands the robot to position its hand. The robot moves and lowers its hand to grasp the part. If it fails, the robot assumes that it accidentally bumped and moved the part, and it must therefore take another set of measurements and try again until it successfully grasps the part.

In practice, several factors affect the robot's grasping performance besides the accuracy of the localization system. In particular, the gripper's quality is highly important [10]. Grippers of different materials trade off between ease of grasping and stability of holding a part. Firmer surfaces provide rigid control, but may fail to pick up the part due to inadequate friction between the grippers and the part. The size of the gripper is another determining factor. Today's commercial manipulator robots have a typical hand size of >10 cm [42]. The KUKA youBot designed for educational and research use has a slightly smaller hand (its gripper aperture is 7.5 cm).

In order to isolate the impact of RF-Compass on the grasping task, we perform this evaluation using trace-driven simulations. We simulate the IKEA leg grasping procedure by feeding to the simulator the empirical results in §6.2, which measure RF-Compass's error in locating the center of the table leg and identifying its orientation. For each grasp attempt, we draw the position and orientation error uniformly at random from the empirical results of our experiments. We repeat the simulation for different robot hand sizes.

Fig. 14 shows the results for the case where two tags on the part are used. Each CDF contains 1000 runs of the grasping task where the part width is 5 cm. We vary the gripper size among 7.5 cm, 8.5 cm and 10 cm. If the robot's gripper is 7.5 cm wide, and the IKEA part has only two RFID tags, the robot needs to try on average 9 times to grasp the part. If the robot's gripper is 8.5 cm wide, the average number of trials decreases to 5. With a hand size of 10 cm, the robot can grasp the part on the first try in over 80% of the time.

Fig. 15 shows the equivalent CDFs for the case where four tags are deployed on the IKEA part. Now, even with a small 7.5 cm gripper like that of the KUKA youBot, the robot can immediately grasp the object almost 90% of the time.

In conclusion, the accuracy achieved by RF-Compass can enable a commercial manipulator robot with a typical hand size of >10 cm to reliably grasp the furniture part with 2 RFIDs. Robots with a smaller gripper, e.g., a 7.5 cm wide gripper, can also be supported if four RFID tags are placed on the target object.

7. CONCLUSION

This paper presents RF-Compass, an RFID-based system for robot object manipulation. RF-Compass's key innovation is an iterative localization algorithm that leverages the robot's consecutive moves to iteratively refine its estimates of the desired object's location and orientation. It can pinpoint the center position and the orientation to an accuracy of a few centimeters and degrees respectively, providing the necessary precision for object grasping and manipulation using a purely RF-based solution.

By bridging robotics and advances in wireless signal processing, RF-Compass delivers a highly accurate robot object manipulation scheme which opens up a wide range of exciting opportunities. We believe future systems can leverage the techniques in this paper to enable new features in robotic applications such as picking up an
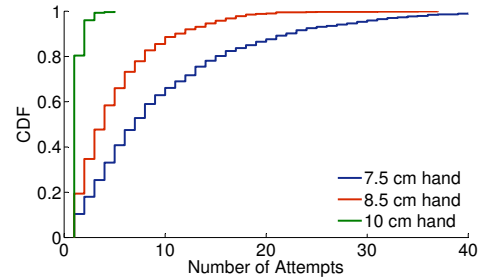

Figure 14—CDF of the number of attempts made by the robot before succeeding at grasping the part with 2 RFIDs: With 2 tags on the furniture part, a commercial manipulator robot with a typical hand size of >10 cm can grasp the part successfully upon the first try in over 80% of the cases.
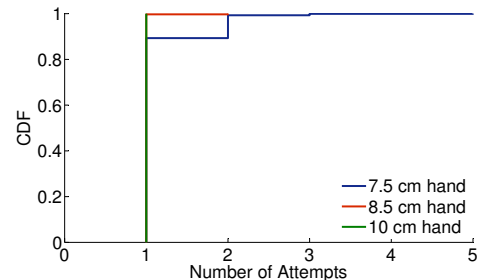

Figure 15—CDF of the number of attempts made by the robot before succeeding at grasping the part with 4 RFIDs: With 4 tags on the furniture part, robots with a small hand 7.5 cm such as the youBot can retrieve the part successfully upon the first try in almost 90% of the cases.

object from a pile, grasping objects with arbitrary shapes, complex assembly tasks, robot collaboration, and many others.

8. REFERENCES

[1] F. Adib and D. Katabi. See through walls with Wi-Fi! In *ACM SIGCOMM*, 2013.

[2] Alien Technology Inc. ALN-9640 Squiggle Inlay. www.alientechnology.com.

[3] L. Atzori, A. Iera, and G. Morabito. The internet of things: A survey. *Computer Networks*, 2010.

[4] S. Azzouzi et al. New measurement results for the localization of UHF RFID transponders using an Angle of Arrival (AoA) approach. In *IEEE RFID*, 2011.

[5] P. Bahl and V. Padmanabhan. RADAR: an in-building RF-based user location and tracking system . In *IEEE INFOCOM*, 2000.

[6] J. Biswas and M. Veloso. Wifi localization and navigation for autonomous indoor mobile robots. In *IEEE ICRA*, 2010.

[7] M. Bonert, L. Shu, and B. Benhabib. Motion planning for multi-robot assembly systems. *International Journal of Computer Integrated Manufacturing*, 2000.

[8] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[9] M. Buettner and D. Wetherall. A software radio-based uhf rfid reader for phy/mac experimentation. *IEEE RFID*, 2011.

[10] I. Bullock, R. Ma, and A. Dollar. A hand-centric classification of human and robot dexterous manipulation. *IEEE Trans. on Haptics*, 2013.

[11] H. Chae and K. Han. Combination of RFID and vision for mobile robot localization. In *IEEE ICSSINP*, 2005.

[12] G. Champleboux, S. Lavallee, R. Szeliski, and L. Brunie. From accurate range imaging sensor calibration to accurate model-based 3d object localization. In *IEEE CVPR*, 1992.

[13] R. T. Chin and C. R. Dyer. Model-based recognition in robot vision. *ACM Computing Surveys (CSUR)*, 1986.

[14] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan. Indoor localization without the pain. In *ACM MobiCom*, 2010.

[15] A. Collet, D. Berenson, S. S. Srinivasa, and D. Ferguson. Object recognition and full pose registration from a single image for robotic manipulation. In *IEEE ICRA*, 2009.

[16] EPCglobal Inc. EPCglobal Class 1 Generation 2.

[17] J. Fitch. Synthetic aperture radar. 1988.

[18] Frost & Sullivan. Global RFID healthcare and pharmaceutical market. Industry Report, 2011.

[19] Frost & Sullivan. Global RFID market. Industry Report, 2011.

[20] M. Grant, S. Boyd, and Y. Ye. CVX: Matlab software for disciplined convex programming. http://cvxr.com/cvx.

[21] W. E. L. Grimson and D. Huttenlocher. On the verification of hypothesized matches in model-based recognition. *Computer Vision - ECCV*, 1990.

[22] D. Hahnel, W. Burgard, D. Fox, K. Fishkin, and M. Philipose. Mapping and localization with RFID technology. In *IEEE ICRA*, 2004.

[23] S. Han, H. Lim, and J. Lee. An efficient localization scheme for a differential-driving mobile robot based on RFID system. *IEEE Trans. Industrial Electronics*, 2007.

[24] Ikea lack table. www.ikea.com/us/en/catalog/products/20011413.

[25] E. Inc. Universal software radio peripheral. http://ettus.com.

[26] S. Jorg, J. Langwald, J. Stelter, G. Hirzinger, and C. Natale. Flexible robot-assembly using a multi-sensory approach. In *IEEE ICRA*, 2000.

[27] K. Joshi, S. Hong, and S. Katti. Pinpoint: Localizing interfering radios. In *Usenix NSDI*, 2013.

[28] K. Khoshelham and S. O. Elberink. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 2012.

[29] M. Kim and N. Y. Chong. Rfid-based mobile robot guidance to a stationary target. *Mechatronics*, 2007.

[30] M. Kim and N. Y. Chong. Direction sensing RFID reader for mobile robot navigation. *IEEE Trans. Automation Science and Engineering*, 2009.

[31] R. A. Knepper, T. Layton, J. Romanishin, and D. Rus. Ikeabot: An autonomous multi-robot coordinated furniture assembly system. In *IEEE ICRA*, 2013.

[32] KUKA, youBot specification sheet. youbot-store.com/downloads/ProductFlyer_KUKA_youBot_2pages.pdf.

[33] Laird Technologies. Crushcraft S9028PCRW RFID antenna.

[34] M. T. Mason. Creation myths: The beginnings of robotics research. *IEEE Robotics Automation Magazine*, 2012.

[35] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar. The GRASP multiple micro-UAV testbed. *IEEE Robotics & Automation Magazine*, 2010.

[36] L. M. Ni et al. Landmarc: indoor location sensing using active rfid. *Wirel. Netw.*, 2004.

[37] P. Nikitin et al. Phase based spatial identification of uhf rfid tags. In *IEEE RFID*, 2010.

[38] S. Nirjon and J. Stankovic. Kinsight: Localizing and tracking household objects using depth-camera sensors. In *IEEE DCOSS*, 2012.

[39] S. Park and S. Hashimoto. Autonomous mobile robot navigation using passive RFID in indoor environment. *IEEE Trans. Industrial Electronics*, 2009.

[40] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng. ROS: an open-source robot operating system. In *IEEE ICRA Workshop on Open Source Robotics*, 2009.

[41] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen. Zee: zero-effort crowdsourcing for indoor localization. In *ACM Mobicom*, 2012.

[42] Rethink robotics, Baxter specification sheet. www.rethinkrobotics.com/files/8513/4792/6562/ProductBrochure_WebDownload.pdf.

[43] J. Rifkin and E. Kruger. *The end of work*. 1996.

[44] S. Schneegans, P. Vorst, and A. Zell. Using RFID snapshots for mobile robot self-localization. In *ECMR*, 2007.

[45] A. Smith et al. Tracking Moving Devices with the Cricket Location System. In *ACM MobiSys*, 2004.

[46] ThingMagic. Why Use RFID. www.thingmagic.com/.

[47] VICON T-Series. www.vicon.com/products/documents/Tseries.pdf.

[48] J. Wang, H. Hassanieh, D. Katabi, and P. Indyk. Efficient and Reliable Low-Power Backscatter Networks. In *ACM SIGCOMM*, 2012.

[49] J. Wang and D. Katabi. Dude, Where's my card? RFID Positioning That Works with Multipath and Non-Line of Sight. In *ACM SIGCOMM*, 2013.

[50] J. Xiong and K. Jamieson. Arraytrack: A fine-grained indoor location system. In *Usenix NSDI*, 2013.

[51] Y. Ye. Improved complexity results on solving real-number linear feasibility problems. *Math. Program.*, 2006.

[52] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 2006.

[53] T. Zimmerman. Assessing the capabilities of RFID technologies. Gartner, 2009.