Autonomous Robots (2023) 47:1137–1154
https://doi.org/10.1007/s10514-023-10137-1

FuseBot: mechanical search of rigid and deformable objects via multi-modal perception

Tara Boroushaki[1] · Laura Dodds[1] · Nazish Naeem[1] · Fadel Adib[1]

Received: 13 February 2023 / Accepted: 20 August 2023 / Published online: 23 September 2023

Abstract

Mechanical search is a robotic problem where a robot needs to retrieve a target item that is partially or fully-occluded from its camera. State-of-the-art approaches for mechanical search either require an expensive search process to find the target item, or they require the item to be tagged with a radio frequency identification tag (e.g., RFID), making their approach beneficial only to tagged items in the environment. We present FuseBot, the first robotic system for RF-Visual mechanical search that enables efficient retrieval of both RF-tagged and untagged items in a pile. Rather than requiring all target items in a pile to be RF-tagged, FuseBot leverages the mere existence of an RF-tagged item in the pile to benefit both tagged and untagged items. Our design introduces two key innovations. The first is *RF-Visual Mapping*, a technique that identifies and locates RF-tagged items in a pile and uses this information to construct an RF-Visual occupancy distribution map. The second is *RF-Visual Extraction*, a policy formulated as an optimization problem that minimizes the number of actions required to extract the target object by accounting for the probabilistic occupancy distribution, the expected grasp quality, and the expected information gain from future actions. We built a real-time end-to-end prototype of our system on a UR5e robotic arm with in-hand vision and RF perception modules. We conducted over 200 real-world experimental trials to evaluate FuseBot and compare its performance to a state-of-the-art vision-based system named X-Ray (Danielczuk et al., in: 2020 IEEE/RSJ international conference on intelligent robots and systems (IROS), IEEE, 2020). Our experimental results demonstrate that FuseBot outperforms X-Ray's efficiency by more than 40% in terms of the number of actions required for successful mechanical search. Furthermore, in comparison to X-Ray's success rate of 84%, FuseBot achieves a success rate of 95% in retrieving untagged items, demonstrating for the first time that the benefits of RF perception extend beyond tagged objects in the mechanical search problem.

Keywords Mechanical search · RF-visual perception · RF-visual fusion · RFID · Robotic grasping

1 Introduction

There has been increasing interest in robotic systems that can find and retrieve occluded items in unstructured environments such as warehouses, retail stores, homes, and manufacturing (Danielczuk et al., 2019, 2020; Boroushaki et al., 2021a, b; Huang et al., 2020). For example, in e-commerce warehouses, there is a need for robots that can package customer orders from unsorted inventory or process returns from a miscellaneous pile. Similarly, in manufacturing plants, robots need to find and retrieve specific tools from the environment (e.g., a wrench) that they need for assembly tasks. In many of these scenarios, the target item may be partially or fully occluded from the robot's camera, requiring the robot to actively explore the entire environment to find and retrieve the desired item.

Existing robotic systems that aim to address this *mechanical search* problem broadly fall in two main categories. The first relies entirely on vision-based perception (Danielczuk et al., 2019, 2020; Huang et al., 2020). In these systems, the robot typically performs active perception by moving its camera around a pile to identify the target item through partial occlusions, and/or it performs manipulation to declutter the scene by removing occluding items until it can observe

✉ Tara Boroushaki
tarab@mit.edu

Laura Dodds
ldodds@mit.edu

Nazish Naeem
nazishn@mit.edu

Fadel Adib
fadel@mit.edu

[1] Massachusetts Institute of Technology, Cambridge, USA

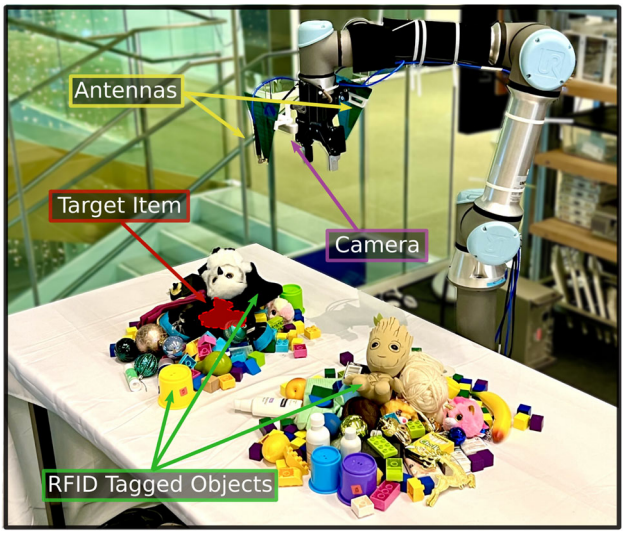


Fig. 1 RF-visual mechanical search. FuseBot uses RF and visual sensor data (from wrist-mounted camera and antenna) to perform mechanical search and extract the occluded target items from the piles of both RFID tagged and non-tagged items

the target. While this category of systems can perform well on relatively small piles, they become inefficient in complex scenarios with larger or multiple piles. The second category of systems leverages radio frequency (RF) perception in addition to vision-based perception (Boroushaki et al., 2021a, b; Wang et al., 2013). Unlike visible light and infrared, RF signals can go through standard materials like cardboard, wood, and plastic. Thus, recent systems have leveraged RF signals to locate fully occluded objects tagged with widely-deployed, passive, 3-cent RF stickers (called RFIDs). By identifying and locating the RFID-tagged target items through occlusions, these systems can make the mechanical search process much more efficient. However, the benefits of existing systems in this category are restricted to scenarios where all target items are tagged, thus providing limited benefit in more common scenarios where only a subset of items are tagged with RFIDs.

In this paper, we ask the following question: Can we design a robotic system that performs efficient RF-Visual mechanical search for both RF-tagged and non-tagged target objects? Specifically, rather than requiring all items to be RF-tagged, we consider more realistic and practical scenarios where only a subset of items are tagged, and ask whether one can improve the efficiency of retrieving non-tagged target items by leveraging RF perception. A positive answer to this question would extend the benefits of RF perception to new application scenarios, such as those where the target item cannot be tagged with inexpensive RFIDs (e.g., metal tools and liquid bottles)[1] and instances when the robot is presented with piles of items that are not fully tagged.

We present ***FuseBot***, a robotic system that can efficiently find and extract tagged and non-tagged items in line-of-sight, non-line-of-sight, and fully occluded settings. Similar to past work that leverages RF perception, FuseBot uses RF signals to identify and locate RFID tags in the environment with centimeter-scale precision. Unlike the past systems, it can efficiently extract both non-tagged and tagged items that are fully occluded. As shown in Fig. 1, FuseBot integrates a camera and an antenna into its robotic arm and leverages the robot movements to locate RFIDs, model unknown/occluded regions in the environment, and efficiently extract target items from under a pile independent of whether or not they are tagged with RFIDs.

The key intuition underlying FuseBot's operation is that knowing where an RFID-tagged item is within a pile provides useful information about the pile's occupancy distribution and allows the robot to significantly narrow down the candidate locations of non-tagged items. In its simplest form, knowledge of where an RFID-tagged item is within a pile negates the possibility of another item occupying the same location. Since the in-hand antenna allows the robot to localize all RFID tags in a pile, the robot can leverage this knowledge to narrow down the likely locations of a non-tagged target item, and thus plan efficient retrieval policies for these items.

Translating this high-level idea into a practical system is challenging. While the in-hand antenna can locate each RFID as a single point in 3D space, it cannot recover the 3D volumetric occupancy map of the object an RFID is attached to. Since an RFID is attached to the object's surface and not at its center, there is uncertainty about both the position and orientation of the tagged item. The problem is further complicated by the fact that retrieving an occluded item involves manipulating the environment (e.g., by removing occluding objects to uncover the target). Here, uncertainty about the target object's location makes it difficult to identify the optimal manipulation actions to most efficiently reveal and extract the target.

FuseBot introduces two key components that together allow it to overcome the above challenges:

(a) RF-Visual Mapping FuseBot's first component constructs a probabilistic occupancy map of the target item's location in the pile by fusing information from the robot's in-hand camera and RF antenna as shown in Fig. 2a. This component localizes the RFIDs in the pile and applies a conditional (shape-aware) RF kernel to construct a negative 3D probability mask, as shown in the red regions of Fig. 2b.

[1] It is worth noting certain RFIDs can work on metal and liquids, but are much more expensive than the 3-cent passive RFIDs, making prohibitive for widespread adoption.

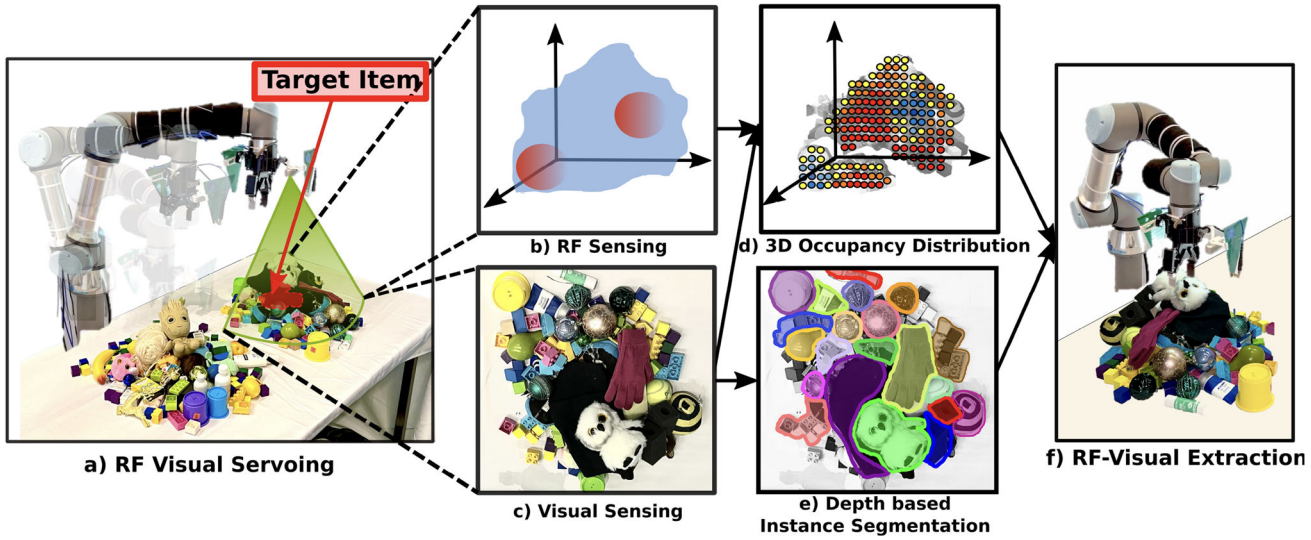


Fig. 2 RF-visual mapping and RF-visual extraction. **a** As FuseBot moves, it observes the environment using the wrist mounted camera and RF module. **b** Using the RF measurements, FuseBot localizes the RFID tagged items in the environment and computes RF kernels. **c** Using the wrist mounted camera, FuseBot observes the environment. **d** FuseBot fuses the vision observations and the RF kernels to create a 3D occupancy distribution map which is visualized as a heat map. **e** FuseBot performs instance segmentation of the objects in the environment using the depth information from the camera. **f** FuseBot optimized its extraction strategy by integrating the 3D occupancy distribution over each of the object segments and efficiently retrieves the target

By combining this information with its visual observation of the 3D pile geometry (shown Fig. 2c), as well as prior knowledge of the target object's geometry, FuseBot creates a 3D occupancy distribution, shown as a heatmap in Fig. 2d, where red indicates high probability and blue indicates low probability for the target item's location. In this example, it is worth noting how the probability of the occluded target item is lower near the locations of RFID-tagged objects. Section 4 describes this component in detail, and how it also leverages the geometry of the tagged items and the pile.

(b) RF-Visual Extraction Policy After computing the 3D occupancy distribution, FuseBot needs an efficient extraction policy to retrieve the target item. Extraction is a multi-step process that involves removing occluding items and iteratively updating the occupancy distribution map. To optimize this process, we formulate extraction as a minimization problem over the expected number of actions that takes into account the expected information gain, the expected grasp success, and the probability distribution map. To efficiently solve this problem, FuseBot performs depth-based instance segmentation, as shown in Fig. 2e. The segmentation allows it to integrate the 3D occupancy distribution over each of the object segments, and identify the optimal next-best-grasp, as we describe in detail in Sect. 5.

We implemented a real-time end-to-end prototype of FuseBot with a Universal Robot UR5e (Universal Robots, 2021) and Robotiq 2f-85 gripper (Robotiq, 2019). As shown in Fig. 1, we mount an Intel RealSense Depth camera D415 (Intel RealSense, 2019) and log-periodic antennas on the wrist of the robotic arm. Our implementation localizes the RFIDs by processing measurement obtained from the log-periodic antennas using BladeRF software radios (Nuand, 2021).

We ran over 200 real-world experimental trials to evaluate FuseBot. We compared our system to a state-of-the-art system called X-Ray (Danielczuk et al., 2020), which computes a 2D occupancy distribution based on an RGB-D image. Our evaluation demonstrates the following:

- FuseBot can efficiently retrieve complex, non-tagged items in line-of-sight and fully occluded settings, across different target objects and number of RFID tags. It succeeds in 95% of trials across a variety of scenarios, while X-Ray was able to extract the target item in 84% of the scenarios.
- In scenarios where FuseBot and X-Ray succeed in mechanical search, FuseBot improves the efficiency of extraction by more than 40%. Specifically, it reduces the number of actions needed for successful retrieval from 5 to 3 actions in the median, and from 11 to 6 in the 90th percentile.
- Our results also demonstrate that the efficiency gains from FuseBot's RF-Visual mechanical search increase with the number of tagged items in the environment, reaching as much as 2.5× improvement over X-Ray in environments where 25% of (non-target) items are RF-tagged and 4× improvement when the target item is tagged.

Contributions FuseBot is the first system that enables mechanical search and extraction of both non-tagged and tagged RFID items in non-line-of-sight and fully-occluded settings. The system introduces two new primitives, *RF-Visual Mapping* and *RF-Visual Extraction*, to enable RF-Visual scene understanding and efficient retrieval of target items. The paper also contributes a real-time end-to-end prototype implementation of FuseBot, and an evaluation that demonstrates the system's practicality, efficiency, and success rate in challenging real-world environments.

2 Related work

Interest in the problem of mechanical search dates back to research that recognizes objects through or around partial occlusions via active and interactive perception. Researchers explored the use of perceptual completion to identify partially occluded objects (Huang et al., 2012; Price et al., 2019), and developed systems that perform active perception whereby a robot moves a camera around the environment in order to search for items that are partially visible (Aydemir et al., 2011; Bajcsy, 1988; Bohg et al., 2017). Other areas of research focused on efficiently grasping partially occluded objects using physics-based planners (Dogar et al., 2012). While these works made significant progress on the task of finding and retrieving partially occluded objects, they do not extend to mechanical search scenarios where the target object is fully occluded.

Over the past few years, there has been rising interest in the mechanical search problem for fully occluded objects, whereby the robot actively manipulates the environment to uncover target objects. The majority of systems for mechanical search rely entirely on vision, and employ heuristics or knowledge of the pile structure in order to inform the search process. For example, recognizing that mechanical search is a multi-step retrieval process, pioneering research in this space used a heuristic-based approach to remove larger items in the environment to uncover the largest area and maximize information gain at each step (Danielczuk et al., 2019). More recent work has started looking at the structure of the pile and constrains the potential target item locations by leveraging the geometry of both the pile and the target object (Danielczuk et al., 2020). Other work has also looked at lateral search, where objects are retrieved from the side rather than from a pile (Huang et al., 2020; Avigal et al., 2021). One of the main challenges of this vision-based approach to mechanical search is that as piles become larger and more complex, the uncertainty grows and the systems become more inefficient. FuseBot builds on this type of research to perform efficient mechanical search of fully-occluded objects, and outperforms state-of-the-art past vision-based systems (as we demonstrate empirically in Sect. 7) especially in the presence of any RFID tagged item.

Most recently, researchers have explored the use of RF perception to address the mechanical search problem (Boroushaki et al., 2021a, b; Wang et al., 2013). This research was motivated by recent advances in RF localization, which has enabled locating cheap, passive, widely-deployed RF-tags (called RFIDs tags) with centimeter scale accuracy, even through occlusions (Ma et al., 2017; Wang & Katabi, 2013; Luo et al., 2019). Thus, by tagging the target object with an RFID, researchers have demonstrated the potential to perform efficient mechanical search by directly locating the target RFID-tagged item in a pile, bypassing the exhaustive search altogether. However, these past systems require the target item to be tagged with an RFID to enable efficient mechanical search and retrieval. Our work is motivated by this line of work, and is the first to bring the benefits of RF perception to non-tagged target items, leveraging the mere existence of RFID tagged items in the pile.

3 System overview

We consider a general mechanical search problem where a robot is tasked with retrieving a target item from a pile. The target item may be unoccluded, partially occluded, or fully occluded from the robot's camera.

We focus on scenarios where one or more items in the pile are tagged with UHF RFID (Radio Frequency IDentification) tags, but where the target item does not need to be tagged with an RFID. We assume that the robot knows the shape of the tagged item, and has a database with the shapes of all RFID-tagged items. Such a database may be provided by the item's manufacturer. The robot is a 6-DOF manipulator with a camera and an antenna mounted on its wrist, and we assume that the target item is kinematically reachable from the robotic arm on a fixed base.

FuseBot's objective is to extract the target(s) from the environment using the smallest number of actions. It starts by using its wrist-mounted antenna to wirelessly identify and locate all RFIDs in the pile, even if they are in non-line-of-sight. Using the RFID locations and its visual observation of the pile geometry, it performs RF-Visual mechanical search in two key steps. The first is *RF-Visual Pile Mapping*, where FuseBot creates a 3D probability distribution of the target object's location within the pile. The second is *RF-Visual Extraction*, where the robot uses the probability distribution and its scene understanding to perform the next-best grasp. The next two sections describe these steps in detail.

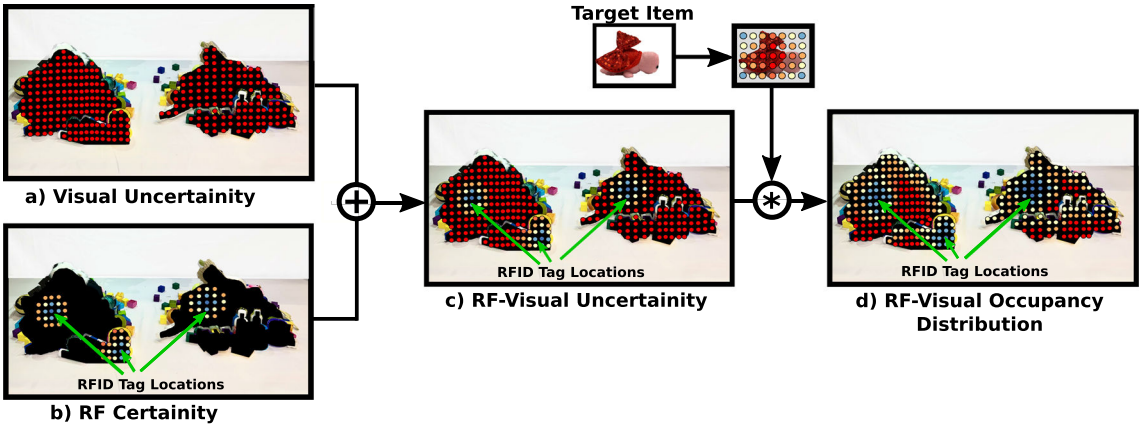


Fig. 3 RF-visual mapping. FuseBot **a** constructs an initial map of unknown regions using visual RGB-D information and **b** uses RFID tag locations to construct RF kernels. **c** It then combines the RF and Visual information to more accurately map probable target locations. **d** Finally, it uses the target object geometry to further refine the probable target locations

4 RF-visual pile mapping

In this section, we explain how FuseBot creates a 3D occupancy distribution of a target item's location in a pile. The process of RF-Visual mapping consists of four key steps where the robot first constructs separate RF and visual maps, then fuses them together, and finally folds in information about the target object's geometry. For clarity of exposition, we focus our discussion on scenarios where the target item is both occluded and non-tagged, and discuss at the end of the section how this technique generalizes to unoccluded and/or non-tagged items.

4.1 Visual uncertainty map

The first step of RF-Visual pile mapping involves constructing a 3D visual uncertainty map of the environment. This map is important to identify all candidate locations of an occluded object. To create the visual uncertainty map, the robot moves its downward pointing wrist-mounted camera above the pile to cover the workspace. It follows a simple square-based trajectory in a plane parallel to the table with a pile, similar to past work that constructs point clouds of piles (Boroushaki et al., 2021b).

FuseBot combines the visual information obtained during its trajectory using an Octomap structure (Hornung et al., 2013). The structure represents the 3D workspace as a voxel grid.[2] Using depth information and the position of the camera, FuseBot can determine whether each voxel in the environment is visible to the camera (the surface of the pile and table), or free space (the air), or occluded (e.g., under the pile or table). Formally, it creates a 3D uncertainty matrix $\mathbf{C}(x, y, z)$ as follows:

$$\mathbf{C}(x, y, z) = \begin{cases} 1 & \text{unobserved voxel} \\ 0 & \text{observed voxel} \end{cases}$$

Here, the higher value (i.e., 1) represents more uncertainty. It is worth noting that, in this representation, both unexplored and occluded regions are considered uncertain.

As an example, consider the sample scenario shown in Fig. 1. This scenario consists of two piles with three RFID-tagged items, and where the target item is a toy (stuffed red turtle shown in the top center) hidden under the pile. The visual uncertainty map is depicted as a heatmap in Fig. 3a. Here, we can see that the regions under the surface of the piles have a high probability (red) of containing the target object.

4.2 RF localization

So far, we have explained how FuseBot constructs a 3D uncertainty map based on the camera's depth information. Next, we explain how it accurately localizes RFIDs to gain more information about the environment. For simplicity, we first describe the localization of a single tag, then describe how we support multiple tags. Our localization system follows three steps:

Step (1) Measuring RFID response First, recall that FuseBot has a wrist-mounted antenna which it uses to perform RF perception. The antenna is used to read and localize RFID tags in the pile. When the antenna transmits radio frequency signals, passive RFID tags harvest energy from this signal to power up and respond with their own identifier. FuseBot then uses these responses to estimate the channel, which contains

[2] In our implementation, each voxel is a $2.5 \times 2.5 \times 2.5$cm cubic volume.

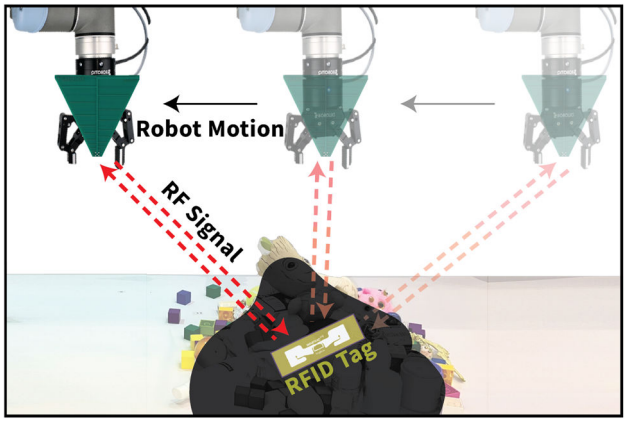


Fig. 4 RF localization. FuseBot sends and receives RF signals (red arrows) to and from the battery-free RFID tag (in yellow) at different vantage points in order to localize the RFID tags in the environment

information about the distance to the tag. We refer readers to Tse and Viswanath (2005) for more details on RF channels.

Formally, if an RFID transmits a signal x(t), and the received signal is y(t), one can estimate the wireless channel $\hat{h}(f_i)$ as:

$$\hat{h}(f_i) = \sum_t y(t)x^*(t)$$

The above describes the channel estimation at a single frequency f_i. FuseBot repeats this process at multiple frequencies to obtain $\{\hat{h}(f_i)\}_i$

Step (2) Leveraging robot mobility for localization Since channel measurements from a single location are not enough to localize an RFID tag in 3D space, FuseBot leverages robotic mobility to collect measurements from different vantage points and combines them to localize the tag. Since FuseBot already requires a scan of the environment to build the Visual Uncertainty map in Sect. 4.1, we leverage this motion and continuously collect RFID channel measurements as the robot moves, allowing us to collect a set of measurements:

$$\{\hat{h}(f_i, p_{a_k})\}_{i,k}$$

where p_{a_k} is the location of the antenna. Figure 4 schematically shows the robot moving and collecting RF measurements in order to localize an RFID tag that is hidden under a pile. The red dotted lines demonstrate the RF signals that are transmitted from the wrist mounted antenna to the RFID tag and then received by the wrist mounted antenna. Remember that unlike visible light, RF signals can traverse through occlusions, and, as a result, the RF channel can be estimated even when the RFID tag is under the pile.

Step (3) Combining measurements Finally, given these measurements, the robot can combine them using a technique called Synthetic Aperture Radar (Curlander & McDonough, 1991). This localization method combines measurements across space and frequency (i.e., $\{\hat{h}(f_i, p_{a_k})\}_{i,k}$) to estimate the probability of the tag being at each point in 3D space. This can be done using the following equation (Curlander & McDonough, 1991):

$$P(p) = \sum_i \sum_k \hat{h}(f_i, p_{a_k}) e^{j2\pi d(p, p_{a_k}) f_i / c} \quad (1)$$

where $P(p)$ is the estimated probability at point p, p_{a_k} is the antennas position at the time of the i^{th} measurement, and $d(p, p_{a_k})$ is the round-trip distance from point p to point p_{a_k}. The final tag location is then estimated to be the location in space with the highest probability:

$$p_{RFID} = \texttt{argmax}(P(p)) \quad (2)$$

where p_{RFID} is the estimated location of the tag.

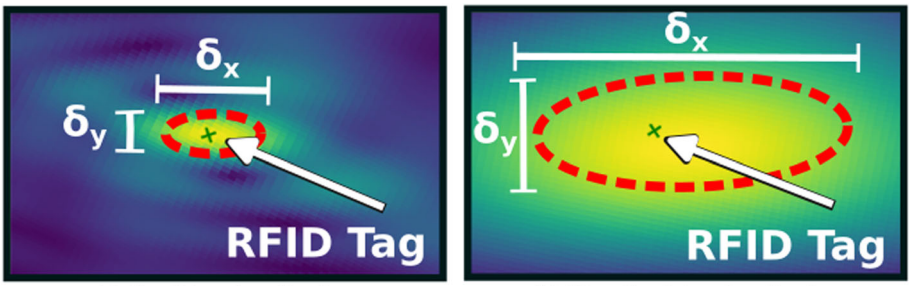
To extend this to any number of RFIDs, we modify step 2 as follows. Instead of continuously reading one RFID, we estimate the channels of all RFID tags in the environment sequentially as the robot is moving.[3] This allows us to collect a set of measurements for each RFID. We then recompute Eqs. 1 and 2 for each RFID in the environment.

Finally, it is worth noting that wireless noise may lead to localization errors. FuseBot's design incorporates a confidence metric (described below) to identify and mitigate such errors. Specifically, if the confidence metric is low, the system can choose either to ignore the corresponding tag altogether or to take more RF measurements that enable it to increase its localization confidence.

To understand whether we have confidently localized an RFID, we leverage information from the probability computed in Eq. 1. For simplicity of exposition, we demonstrate this idea in Fig. 5, which shows a two-dimensional heatmap of the probability, where yellow indicates a higher likelihood of the RFID being located in that location and blue indicates a lower likelihood. We consider two cases. In Fig. 5a), the heatmap shows a small area of high probability surrounding the tag's location (denoted by a green x), which indicates a high level of confidence in the RFID location. On the other hand, Fig. 5b) shows a case where there is a large area of yellow, so FuseBot has a low confidence in the location of the RFID.

To quantify this phenomenon, FuseBot computes the bounding box around the area of the heatmap that is within 0.75dB (∼84%) of the peak value (shown by δ_x and δ_y in Fig. 5). When these dimensions have fallen below a threshold,

[3] RFID readers can read 1000s of tags per second. Moreover, the readers support a medium-access protocol as part of the EPC Gen 2 protocol (http://www.gs1.org/epcrfid/epc-rfid-uhf-air-interface-protocol/2-0-1) that easily allows FuseBot to select specific RFIDs if need be.



(a) Confident localization (b) Non Confident localization

Fig. 5 Confident RFID localization. FuseBot uses the heatmap of the probability (high probability in yellow, low probability in blue) to determine its confidence in an RFID location. **a** A highly confident localization, with a small area of yellow surrounding the RFID tag (green x). **b** A low confidence location, with a large area of yellow (Color figure online)

FuseBot declares the RFID confidently localized. Formally, FuseBot's criteria for declaring a successful RFID localization is:

$$\delta_x < \tau_x \;\&\; \delta_y < \tau_y \;\&\; \delta_z < \tau_z$$

where δ_x, δ_y, and δ_z are the x, y, and z dimensions of the bounding box around the area of the power where $P(p) > 0.84 \max[P(p)]$. τ_x, τ_y, and τ_z are the thresholds in the x, y, and z dimenions, respectively.

4.3 RF certainty map

Next, we explain how FuseBot leverages the estimated RFID locations from the above section to construct a certainty map based on RF measurements.

FuseBot uses the RFID tag locations to identify regions in the pile that the target item is *less* likely to occupy, since they are occupied by the RFID-tagged items (rather than the non-tagged target item). A key challenge here is that the system can only recover the RFID tag's location as a single point in 3D space. Since an RFID is attached to the surface of the tagged item, there remains nontrivial uncertainty about the orientation and exact position of the item in the pile (as it may occupy a non-trivial region in the near vicinity of the localized tag).

RF Kernel FuseBot encodes the uncertainty about the RFID-tagged object's location by constructing a 3D RF kernel that leverages the known dimensions of the tagged object. The RF kernel is modeled as a 3D Gaussian, centered at the RFID tag, and masked with a sphere whose radius is equal to the longest dimension of the tagged item. The spherical mask represents an upper bound on the furthest distance from the tag that the object can occupy. Formally, we represent its RF kernel through the following equation:

$$m(p, p_{RFID}) = \begin{cases} -\frac{e^{\|p-p_{RFID}\|^2/d_s}}{\sqrt{\pi d_s}} & \|p - p_{RFID}\|^2 \le d_l \\ 0 & \|p - p_{RFID}\|^2 > d_l \end{cases}$$

where p is the point where we are evaluating the kernel, p_{RFID} is the location of the RFID, d_s and d_l are the shortest and longest distance of the RFID tagged object's bounding box respectively, and $\|\cdot\|^2$ represents the L2 norm. Here, it is worth noting that the negative sign represents the negative likelihood for the target item to occupy the corresponding region.

In the presence of multiple RFID tagged items, the RF certainty map is a linear combination of all RF kernels

$$\mathbf{R}(x, y, z) = -\sum_{i=0}^{N} m(p, p_i)$$

where N is the number of RFID tagged items in the environment. p_i is the ith RFID location, and $m(p, p_i)$ is the ith RF kernel.

The RF certainty distribution for the example scenario (described in Fig. 1) is shown in Fig. 3b. Since there are three RFID-tagged items in the pile, the figure shows three spherical regions that represent the Gaussians centered at each of the localized RFIDs.

RF-Visual Uncertainty Map: Given both the visual uncertainty map and the RF certainty map, FuseBot constructs an RF-Visual uncertainty map by adding the two maps pixel-wise (i.e., **C** + **R**). In the above example with two piles and three RFID-tagged items, Fig. 3c shows the resulting RF-Visual uncertainty map. Notice how by applying the RF masks as a negative mask to the voxel grid values, FuseBot folded the certainty gained from RF into the uncertainty from the visual information.

4.4 RF-visual occupancy distribution map

So far, we have described how FuseBot constructs a 3D probability distribution of possible locations of the target item by fusing RF and visual information. Next, we describe how FuseBot also leverages the target item's size and shape to further improve the occupancy distribution map. Intuitively, the target's size constrains the potential regions it can occupy in the occluded region since, for example, larger targets cannot fit into narrow regions of the pile.

To fold the target size into the distribution, FuseBot employs a similar approach to the RF kernel described in

Sect. 4.3. Specifically, it creates a target occupancy kernel that summarizes all the possible orientations of a target object using the following target gaussian kernel:

$$k(p) = \begin{cases} \frac{e^{\|p\|^2/(2d_s^2)}}{d_s\sqrt{2\pi}} & \|p\|^2 \leq \frac{d_l}{2} \\ 0 & \|p\|^2 > \frac{d_l}{2} \end{cases} \quad (3)$$

where p is the point where we are evaluating the kernel, d_s and d_l are the shortest and longest distance of the target object bounding box respectively, and $\|\cdot\|^2$ represents the L2 norm.[4]

To combine the geometric data from this target gaussian kernel with the previously computed RF-Visual uncertainty map, FuseBot performs a 3D convolution of the RF-Visual uncertainty map and the target's gaussian kernel. Intuitively, after convolution, the regions that can fit the item of interest in more possible orientations will have voxels with higher weights than other regions of the unknown environment. Hence, the resulting 3D occupancy distribution now encodes the visual uncertainty, RFID tagged items, and the shape and size of the target item.

Figure 3d shows the resulting RF-Visual occupancy distribution from this convolution operation (for the scenario described earlier in Fig. 1). Notice that in this distribution, regions near the RFID tags, as well as those near the edge of the pile, have lower probabilities (blue/white) than other regions in the pile.

4.5 Generalizing to other scenarios

Our discussion so far has focused on the case of a fully-occluded non-tagged target item. The method can be generalized to other scenarios in a number of ways:

4.5.1 Tagged target object

In scenarios where the target object is tagged with an RFID tag and is not in the line of sight, FuseBot uses the calculated RF kernel in order to build the occupancy distribution of the RFID tagged target object. The RF kernel in this case is positive and the visual uncertainty is ignored. FuseBot in this case knows where the target object is and declutters the environments efficiently to extract the target object.

4.5.2 Unoccluded target object

In cases where the target object is unoccluded (or partially occluded), FuseBot can leverage prior approaches for identification and grasping to retrieve the target item from the pile (Chen et al., 2020; Danielczuk et al., 2019; Krizhevsky et al., 2012; Liu & Deng, 2015).

4.5.3 Deformable RFID tagged objects

In principle, FuseBot's probabilistic approach described so far allows it to operate with deformable objects. However, to further improve the efficiency for such objects, we designed a more advanced model. Recall that from the recorded data in the RFID dataset, FuseBot knows if an RFID tagged object is deformable or rigid. Specifically, when a deformable RFID tagged object is present under a pile, it is likely to compress, changing the object's dimensions. This compression causes the object to deviate from the model of the existing RF kernel. FuseBot can leverage this observation to update the RF kernel for such deformable objects. Specifically, instead of using a spherical RF kernel as mentioned in Sect. 4.3, which is more representative of rigid objects whose dimensions are fixed, we introduce a *Deformable RF Kernel*.

We demonstrate this concept in Fig. 6. Figure 6a shows the RFID tagged object before it was deformed. Figure 6b shows the same RFID tagged object under a pile, deformed due to the weight of the rest of the pile. Figure 6c shows the original spherical RF kernel with variance $\sigma = d_s/2$ (as described in 4.3), with blue indicating more negative and red indicating more positive probability. This RF kernel is overlayed with the compressed deformable object that the kernel is attempting to model. In this case, the model poorly aligns with the object. Instead, Fig. 6d shows the new deformable RF kernel. The variances of the Gaussian are updated to create an elliptical kernel, better matching the expected shape of the object.

Formally, we first define a deformation factor for the RFID tagged object, $\alpha(\rho, z) \in [0, 1]$, which estimates how deformed the object is. Here, $\alpha(\rho, z) = 0$ represents a fully deformed object and $\alpha(\rho, z) = 1$ represents a non-deformed object:

$$\alpha(\rho, z) = \begin{cases} 1 & \rho = 1 \\ z/z_{max} & \rho = 0 \end{cases}$$

where $\rho \in \{0, 1\}$ is 1 if the object is rigid and 0 if the object is deformable, z is the height of the RFID location from the table surface, and z_{max} is the maximum height of the pile directly above the RFID tag location.[5] Then, we define the deformable RF kernel as:

[4] One interesting difference between the RF kernel and the target kernel is that the RF kernel is larger since the RFID tag is on the surface of the object, while the target item kernel is defined from the object's center (d_l for the RF kernel vs $d_l/2$ for the target kernel).

[5] In our implementation, z_{max} is the maximum height of the pile within a 3 cm radius of the tag's (x,y) location.

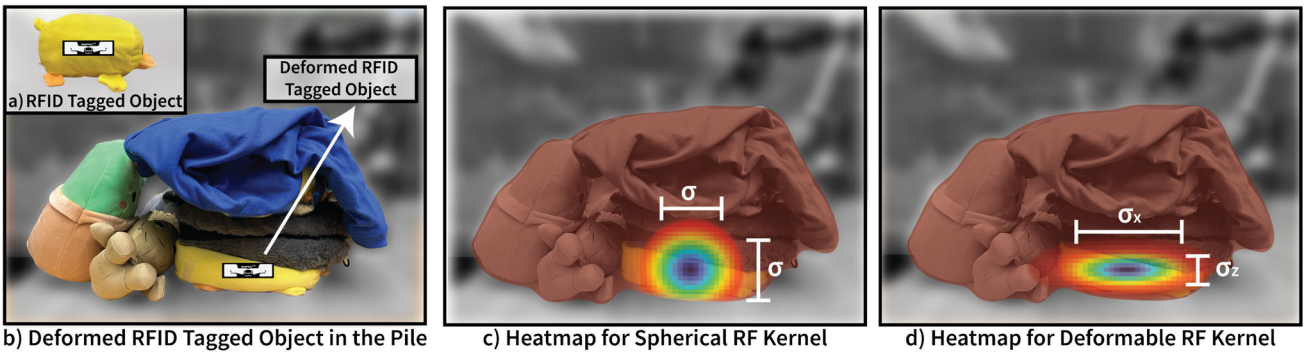


Fig. 6 Spherical and Deformed RF Kernel. **a** A non-deformed RFID tagged object. **b** RFID tagged object that is deformed (compressed in the vertical direction and expanded in the horizontal direction) under a pile of objects. **c** The heatmap of the spherical RF kernel overlayed on the RFID tagged object if deformation is not considered. **d** The heatmap of the deformable RF kernel overlayed on the RFID tagged object when deformation is considered

$$m_d(p, p_{RFID}) = \begin{cases} -\frac{e^{\frac{1}{2}(p-p_{RFID})^T \Sigma^{-1}(p-p_{RFID})}}{\sqrt{2\pi|\Sigma|}} & \|p - p_{RFID}\|^2 \leq d_l \\ 0 & \|p - p_{RFID}\|^2 > d_l \end{cases}$$

$$\Sigma = \begin{bmatrix} \sigma_x & 0 & 0 \\ 0 & \sigma_y & 0 \\ 0 & 0 & \sigma_z \end{bmatrix}$$

where Σ is the covariance matrix, and σ_x, σ_y, and σ_z are the variances in the x, y, and z dimensions, respectively:

$$\sigma_x = \sigma_y = (2 - \alpha(\rho, z))\frac{d_s}{2}$$

$$\sigma_z = \alpha(\rho, z)\frac{d_s}{2}$$

5 RF-visual extraction policy

In the previous section, we explained how FuseBot builds a 3D RF-Visual occupancy distribution for a target item's location. Given this distribution, one might think that the robot could immediately move towards the voxel with the highest probability to extract the target object. However, since the target object is fully occluded, the robot cannot directly access it. Instead, it must first remove anything covering the target object. In this section, we describe FuseBot's RF-Visual extraction policy that decides which object to remove in order to most efficiently extract the target object.

The goal of designing the extraction policy is to minimize the overall number of actions required to retrieve the target object. If the robot was certain of the target item's location, it could simply remove anything covering the object, then extract the target object. However, while FuseBot leverages RF-Visual perception to minimize uncertainty, the occupancy distribution may still have multiple areas of high probability, leaving ambiguity in the target item's location. One could think of moving towards the region with the highest probability and searching for the target object there until it either finds the object or eliminates the search area. However, this may result in an inefficient search, especially in complex scenarios, where there are multiple large piles. Thus, to enable efficient retrieval, FuseBot needs an extraction policy that not only leverages the probability distribution of the target item's location but also the expected information gain of a given action and the likelihood of a successful grasp action.

At the core of enabling an efficient retrieval policy is identifying the next best object to grasp. To this end, FuseBot transform its voxel-based representation of the environment into an object-based representation, which assigns a certain expected gain for grasping each of the visible objects. To do this, FuseBot performs instance segmentation which gives the mask and surface area of each visible object in the scene, as shown in Fig. 7a. Next, in Fig. 7c, it vertically projects all the voxels below a given mask onto the mask and integrates over the mask area. In principle, this provides it with the total utility of extracting the corresponding item (including both the probability distribution and information gain).

Note however that the approach of simply projecting all the probability below an object onto the surface assumes that removing that object would reveal all the voxels below it. In practice, this is not true because the object only has a limited thickness. While FuseBot does not know the thickness of each item, we can safely assume that voxels near the top of the pile are more likely to be eliminated when an object is removed. To bias the search towards this information gain, FuseBot applies a weighting function that increases the weights of voxels closer to the surface of the pile. The sum of these weighted probabilities, or score of each mask, now optimizes for both the information gain and probable tag locations for each visible object. The score is formalized in the below equation:

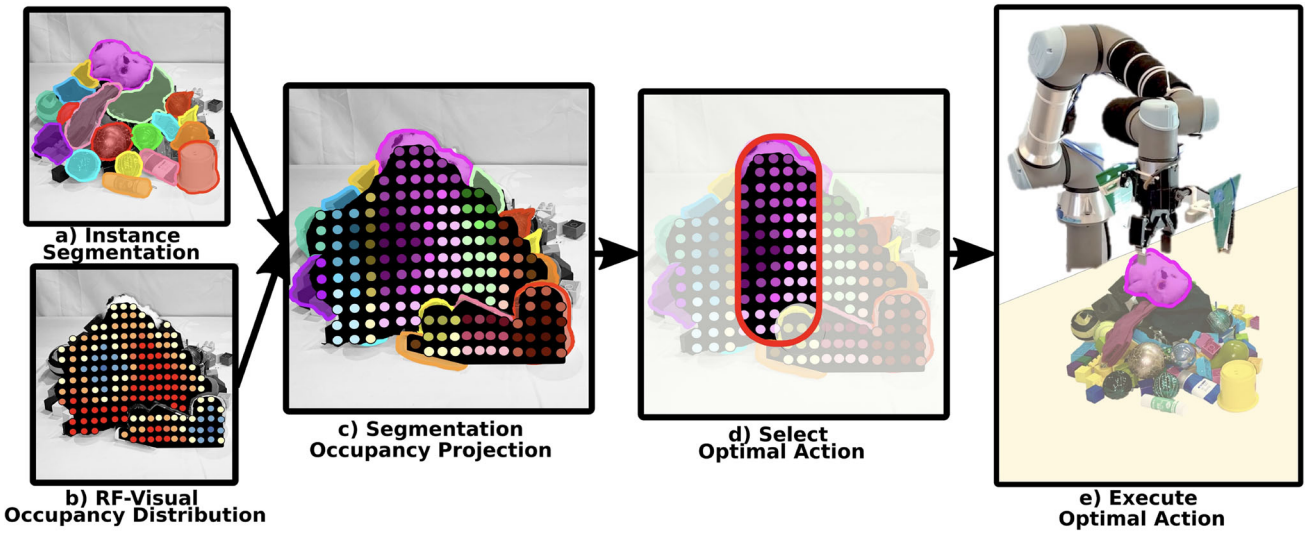


Fig. 7 RF-visual extraction. **a** FuseBot performs depth-based object segmentation to separate different objects in the environment. **b** FuseBot uses the 3D occupancy distribution of the target item. **c** FuseBot projects the occupancy distribution on each segmented mask. **d** FuseBot sums the projected distribution on the area of each mask, and then chooses the mask with the highest sum. **e** FuseBot chooses the next-best-grasp to extract the target item

$$s_i = \sum_{x,y \in m_i} \sum_{z=0}^{z_{m_i}} \gamma^{\frac{(z_{m_i} - z)}{0.025}} \times p_{x,y,z} \tag{4}$$

where s_i is the score of mask i, m_i is all (x,y) points contained within the ith mask, z_{m_i} is the maximum z under the ith mask, and $p_{x,y,z}$ is the probability from the occupancy distribution for point (x,y,z). γ is the discount factor for weighting the probability.[6]

Incorporating Grasp Quality While these scores incentivize both exploiting the probability distribution and maximizing information gain, they do not account for the likelihood of failed grasping attempts. To do this, FuseBot computes the probability of a successful grasp for each point in the environment using a grasp planning network. FuseBot then selects the best possible grasp within each object mask. The grasp qualities of each mask are formalized in the below equation:

$$g_i \leftarrow \max_{(x,y) \in m_i} g(x, y) \tag{5}$$

where g_i is the best grasp probability for the ith mask, $g(x, y)$ is the grasp probability for point (x,y) given by the grasping network, and m_i is all (x,y) points contained within the ith mask.

FuseBot now uses the grasping quality and mask scores to find the optimal extraction policy by optimizing for the following:

$$\max_i \; s_i \times \left\lceil g_i - \tau \right\rceil$$

where i is the mask number and τ is the threshold for acceptable grasping quality. g_i and s_i are the grasping quality and the score for the ith mask, and $\lceil . \rceil$ is the ceiling function. FuseBot first evaluates objects with a greater than τ grasp quality, selecting the object with the best weighted probability score.[7] If no high probability grasps are available, it then selects the object with the best score regardless of grasp quality. The overall algorithm is summarized in Alg. 1.

A few additional points are worth noting:

- Since the workspace may be larger than the field of view of the robot's camera, FuseBot begins by clustering the occupancy distribution and selecting the area with the highest average probability. The robot moves over this area before computing the object masks and grasp qualities and executing the RF-Visual extraction policy. This ensures that FuseBot can extend to any size workspace within the robot arm's reach.
- After each grasp attempt, the robot returns to the position where it grasps in order to locally update the occupancy distribution. It takes new RGB-D images to update a 10cm × 10cm × 10cm region around the grasp point, as well as determine if the target object was uncovered by the latest grasp.

[6] In our implementation, γ is set to 0.95.

[7] In our implementation, τ is set to 0.8.

- At any point, if FuseBot identifies the target object, it ends the RF-Visual extraction policy and proceeds to grasping the target object.

Algorithm 1 RF-Visual Extraction Policy

while Grasp Actions ≤ 15 **do**
  SEGMENTATION
  Compute object segmentation with SDMRCNN(Danielczuk et al., 2019)

  TARGET OBJECT SEARCH
  for mask m_i in SDMRCNN **do**
    if m_i == Target Object **then**
      Grasp Target Object
      Return
    end if
  end for

  MASK SCORING
  for mask m_i in SDMRCNN **do**
    $s_i = \sum_{x,y \in m_i} \sum_{z=0}^{z_{m_i}} \gamma^{\frac{(z_{m_i} - z)}{0.025}} \times p_{x,y,z}$
    $g_i \leftarrow \max_{(x,y) \in m_i} g(x, y)$
  end for

  MASK SELECTION
  if Any $g_i > \tau$ **then**
    $selected\_mask \leftarrow \max_{g_i > \tau} (s_i)$
  else
    $selected\_mask \leftarrow \max_i (s_i)$
  end if
  Grasp $selected\_mask$
end while

6 Implementation

Physical Setup. We implemented FuseBot on a Universal Robots UR5e robot (Universal Robots, 2021) with a Robotiq 2F-85 gripper (Robotiq, 2019). We mounted an Intel Realsense D415 depth camera (Intel RealSense, 2019) and two WA5VJB Log Periodic PCB antennas (850–6500 MHz) (Kent Electronics, 2021) on the gripper. The antennas are connected to two Nuand BladeRF 2.0 Micro software radios (Nuand, 2021) through a Mini-Circuits ZAPD-21-S+ splitter (0.5−2.0 GHz). To obtain RFID locations, we implemented an RFID localization module using the wrist mounted antenna and BladeRFs through a similar method as past work (Ma et al., 2017; Boroushaki et al., 2021b). We used standard off-the-shelf UHF RFID tags (the Smartrac DogBone RFID (Inlay, 2021)) that costs around 3–5 cents.

Control Software The system was developed and tested on Ubuntu 20.04 and ROS Noetic. We used MoveIt [31] as the inverse-kinematic solver to control the robot through the UR Robot Driver package (Universal Robots ROS Driver, 2020).

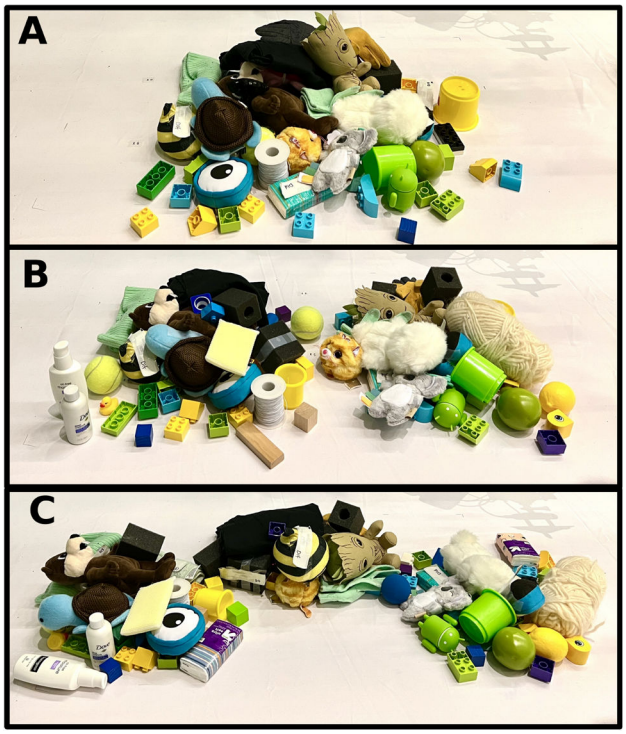


Fig. 8 Example evaluation scenarios. This shows some of the evaluation scenarios for **a** 1 pile, **b** 2 piles, and **c** 3 piles. The target item is fully occluded in all the scenarios

The visual map of the environment is created using Octomap (Hornung et al., 2013). We used Synthetic Depth (SD) Mask R-CNN (Danielczuk et al., 2019) to perform instance segmentation of the scene and segments objects in the scene. To predict the grasping quality from the depth images, we used GG-CNN (Morrison et al., 2018a, b). The baseline, X-Ray (Danielczuk et al., 2020) was implemented based on the published code (Danielczuk et al., 2021).

7 Evaluation

7.1 Real-world evaluation scenarios

We evaluated FuseBot in a variety of real-world scenarios with varying complexity, some of which can be seen in Fig. 8. The scenarios had between 1 and 3 distinct piles of items, 0–10 RFID tagged objects, and a variety of target object and RFID tagged object sizes. Each experiment had one target item and 10–40 other distractor objects. Experiments included varying distances between the target item and the nearest RFID tagged item, including setups with an RFID tagged item touching the target item, RFID tagged items in the same pile as the target item, or all RFID tagged items in different piles than the target item. We also evaluated Fuse-

Bot in scenarios where the target object was tagged with an RFID.[8]

Similar to prior work (Danielczuk et al., 2020) that uses color-based object identification for simplicity, the target item is a red item and FuseBot uses an HSV color segmentation to identify when the target item is in line-of-sight. We note that this step can be replaced by any target template matching network such as the one used in Danielczuk et al. (2019) to identify target objects of any type.

We use everyday objects, both deformable and solid, in our evaluation, including office supplies, toys, and household items like gloves, beanies, tissue packs, travel shampoo, stuffed animals, and thread skeins.

7.2 Baselines

We compared FuseBot's performance with X-Ray (Danielczuk et al., 2020). X-Ray works by estimating 2D occupancy distributions and selecting the object with the highest total probability within its mask to pick up. X-Ray relies entirely on visual information and has no mechanism for RF-perception.

7.3 Metrics

Number of actions We measured the number of grasping actions that were needed to extract the target item from the environment. Actions include grasping a non-target object, target object, or failing to grasp anything.

Success rate We also evaluated the success rate of our system and the baseline. An experimental trial was considered a failure if the robot performed 15 actions and failed to retrieve the target item, or if the robot performed 5 consecutive grasping attempts that failed to grasp any item.

Search and retrieval time We measured the time during which the robot was moving in each successful mechanical search and retrieval task. For FuseBot, this time included the scanning step required to localize the RFIDs.

8 Results

8.1 Baseline comparisons

We evaluated FuseBot and X-Ray in 181 real-world experimental trials. The experiments covered multiple different scenarios of various complexities with 1–3 piles, 0–10 RFID tagged items, and different target object sizes. We tested X-Ray and FuseBot in the exact same scenarios, but we repeated FuseBot multiple times in each scenario with different combinations of RFID tagged item locations and numbers. We measured the number of actions it took to find and retrieve the target item, the success rate of each system, and the search and retrieval time for each system. Recall from Sect. 7(c) that an experimental trial is considered successful if the robot can find and retrieve the target item within 15 actions.

8.1.1 Overall number of actions

Table 1 shows the 10th, 50th, and 90th percentiles of the number of actions required to find and extract the target object. It includes results from FuseBot with RF-tagged target objects, FuseBot with non-tagged target objects, and X-Ray. We make the following remarks:

- FuseBot needs only 3 actions at the median to retrieve non-tagged target item, improving 40% over X-Ray's median number of actions of 5. This shows that FuseBot is able to retrieve non-tagged target items more efficiently than the state-of-the-art vision-based baseline across a variety of scenarios.
- The 90th percentile of FuseBot with non-tagged items is 6 actions, while X-Ray's 90th percentile is 11 actions. This shows that FuseBot is able to perform more reliably, with a 45% improvement over the state-of-the-art at the 90th percentile.
- When searching for a tagged target item, FuseBot requires only 2 actions on median, and 5 actions for the 90th percentile. Note that here it performs better than extracting a non-tagged item. This is expected because localizing the tagged target item reduces the uncertainty about its location and makes mechanical search more efficient. This result shows that FuseBot's performance matches that of past state-of-the-art systems that are designed to extract RFID-tagged items (Boroushaki et al., 2021b)[9]; moreover, unlike these prior systems, FuseBot's benefits also extend to non-tagged items.

8.1.2 End-to-end success rate

Table 1 reports the end-to-end success rate. The results show that FuseBot is able to retrieve the target item 95% of the time for non-tagged and tagged target objects, while X-Ray is only able to do so in 84% of scenarios. This demonstrates that FuseBot not only improves the efficiency, but also the success rate of mechanical search.

[8] Unless otherwise stated, we leverage a spherical RF kernel in our experiments.

[9] See Fig. 14 in Boroushaki et al. (2021b).

Table 1 Efficiency and success rate

System	Number of actions			Success rate (%)
	10th pctl	Median	90th pctl	
FuseBot (untagged)	2	3	6	95
FuseBot (tagged)	2	2	5	95
X-Ray	2	5	11	84

The table shows the success rate as well as the 10th, 50th, and 90th percentiles for the number of actions for both FuseBot and X-ray. The performance of FuseBot is shown for scenarios where the target item is tagged and where it is non-tagged

Table 2 Search and retrieval time

System	Search and retrieval time (s)		
	10th percentile	Median	90th percentile
FuseBot (untagged)	40	62	132
X-ray	50	142	237

The table shows the 10th, 50th, and 90th percentiles for the search and retrieval time of both FuseBot and X-Ray

8.1.3 Search and retrieval time

Table 2 shows the search & retrieval time for both FuseBot and X-Ray. Here, it is worth noting that the robot was programmed to move at the same speed across all experimental trials. We make the following remarks:

- FuseBot only requires 62 s at the median, while X-Ray's median is 142 s, showing more than 2x improvement over the baseline's performance.
- The 90th percentile of FuseBot is 132 s, while X-Ray requires a 90th percentile of 237 s, showing the improvement in reliability of FuseBot over X-Ray.
- This improvement in search & retrieval time shows that FuseBot is more efficient than the baseline despite requiring an additional scanning step.

8.1.4 Scenario complexity

We evaluated FuseBot for non-tagged target objects and X-Ray across three scenarios of different complexities.

- In the first level of complexity, the systems were evaluated on a setup with 2 distinct piles of objects and a total of 20 distractor objects.
- In the second level of complexity, the systems were evaluated on a setup with 3 distinct piles of objects and a total of 25 distractor objects.
- In the third level of complexity, the systems were evaluated on a setup with 3 distinct piles of objects and a total of 42 distractor objects.

Figure 9a plots the number of actions required to find and retrieve the target object for both FuseBot (green) and X-Ray (blue) across three scenarios of different complexities. The error bars indicate the 10th and 90th percentiles. We make the following remarks:

- Across all levels of complexity, FuseBot outperforms the baseline in terms of both its median and 90th percentile efficiency. This shows that the benefits of RF-perception extends to complex scenarios.
- In more complicated scenarios with a larger number of distractor objects, both FuseBot and X-Ray require more actions to retrieve the target item. Interestingly, for more complex scenarios, FuseBot's efficiency gains increase over the baseline.

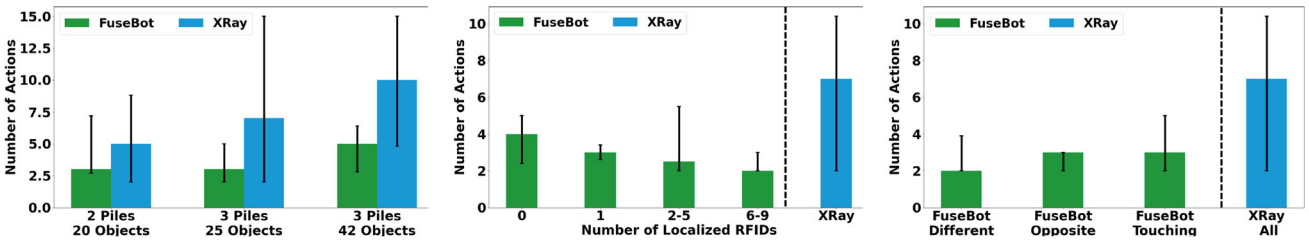
8.2 Microbenchmarks

In addition to baseline comparisons, we performed microbenchmarks to quantify how different factors impact the performance of FuseBot.

8.2.1 Number of RFID tagged items

Recall from 4.3 that FuseBot creates an RF kernel for each identified and localized RFID tagged item, and uses the kernels to build the occupancy distribution. The occupancy distribution gives FuseBot better insight into the location of the target item. We quantified how the system performs with different numbers of RFID tagged items through 54 experiments in the same scenario with varying numbers of RFIDs. In this scenario, we have 3 different piles with a total of 25 objects.

Figure 9b plots the number of actions required to retrieve the target item vs. the number of localized RFIDs in the environment for FuseBot (green) and X-Ray (blue). The error bars denote the 10th and 90th percentiles. Since X-Ray does



(a) **Impact of Scenario Complexity.** (b) **Impact of the Number of RFIDs.** (c) **RFID To Target Item Distance.**

Fig. 9 Impact of different parameters on performance. **a** This figure plots the number of actions required by both FuseBot and X-ray across three different scenarios of increasing complexity. **b** The figure plots the number of actions versus the number of localized RFIDs across fully occluded real-world experiments. **c** This figure plots the median number of actions for FuseBot to retrieve the target item for different RFID to target item distances. X-ray's median number of actions across all scenarios is shown in blue. The error bars denote the 10th and 90th percentile respectively

not utilize RFIDs, the results are not separated by number of RFIDs. We make the following remarks:

- As the number of localized RFIDs in the environment increases, FuseBot's median number of actions decreases, dropping from 4 with no RFIDs to 2 with only 6–9 RFIDs. This improvement in efficiency is expected, because additional RFID tagged items increase the number of RF kernels, which in turn narrows down the candidate locations for the non-tagged target item. More generally, this result shows that leveraging RF perception improves the efficiency of mechanical search, and that the improvement is proportional to the number of RFID tagged items.
- Interestingly, even with 0 RFIDs, FuseBot outperforms X-Ray. Specifically, it requires a median of only 4 actions, while X-Ray requires 7 for the same scenario. This is due to two main reasons. First, while FuseBot leverages a 3D distribution, X-Ray only uses a 2D probability distribution which does not account for the height of different objects. Second, unlike FuseBot, X-Ray does not account for grasp quality when selecting an object to remove from the pile. This makes it susceptible to choosing objects that are more difficult (hence less efficient) to grasp.

8.2.2 Distance from nearest RFID to target item

Our next microbenchmark aims to investigate whether the presence of an RFID-tagged item near the target item would impact the performance. Specifically, one concern with applying the negative mask is that it biases the extraction policy away from the RFID-tagged item. To investigate this, we ran 51 real-world experiments across three scenarios:

- *Touching* In this category, there is at least one RFID tagged item in direct contact with the target item.
- *Opposite Side of Pile* In this category, all RFIDs are either on the opposite side of the target item's pile or in different piles than the target item.
- *Different Piles* In this category, all RFIDs are in different piles than the target object.

Figure 9c plots the median number of actions required to find the target item in each of the three categories of scenarios described above, shown in green. The error bars denote the 10th and 90th percentiles. For comparison, the blue bar show the performance of X-Ray in the same scenario. Since X-Ray does not leverage RFIDs, its performance is not separated into different categories.

We make the following remarks:

- *Different Piles*, *Opposite Side of Pile*, and *Touching* require only 2, 3, and 3 actions at the median, respectively. However, X-Ray requires 7 actions to retrieve the target item. This shows that FuseBot outperforms the baseline across all categories of scenarios, even when an RFID tagged item is touching the target object.
- In *Touching*, the median number of actions is similar to *Different Piles* and *Opposite Side of the Pile*, however the 90th percentile is worse. This is expected because the negative RF mask biases the search away from the target object. However, it is important to note that the 90th is only 5 actions.

8.2.3 Impact of extraction policy

Next, we evaluate the benefits of FuseBot's RF-Visual extraction policy. To do so, we compare to the performance of a naive extraction policy. Unlike FuseBot's policy, this naive policy is designed such that the robot is unaware of the individual objects on the pile, and therefore does not have a way to estimate the expected information gain of removing an item. This naive policy operates in two steps: first, it

Table 3 Impact of extraction policy on efficiency

Extraction policy	Number of actions		
	10th pctl	Median	90th pctl
RF-visual extraction	2.0	2.5	4.0
Naive extraction policy	2.1	4.0	6.9

The table shows the 10th, 50th, and 90th percentiles of the number of actions of FuseBot with different extraction policies

Table 4 Impact of deformable RF kernel on efficiency.

RF kernels	Number of actions		
	10th pctl	Median	90th pctl
Deformable	2.0	3.0	4.0
Spherical	3.0	4.0	6.2

The table shows the 10th, 50th, and 90th percentiles of the number of actions that FuseBot needed to finish the retrieval tasks with deformable RF Kernels and with spherical RF Kernels

selects the voxel with the highest probability in the RF-Visual occupancy distribution (from RF-Visual Mapping); then, it performs the best grasp that is within 5 cm of the voxel's projection on the surface of the pile.

Table 3 shows the 10th, 50th, and 90th percentiles of the number of actions required to successfully extract the target item for FuseBot with both extraction policies for the same set of scenarios with a fully-occluded untagged target item. The result shows that the RF-Visual extraction policy allows FuseBot to successfully complete the task with 2.5 median actions. In contrast, when using the naive extraction policy, it requires 4 median actions. Furthermore, the 90th percentile of FuseBot's extraction policy is only 4 actions, while the naive policy requires 6.9 actions. This performance improvement is due to the fact that FuseBot's RF-Visual extraction policy optimizes for information gain, allowing it to search the environment more efficiently than the simpler extraction policy.

8.2.4 Impact of deformable RF kernel

Recall from Sect. 4.5 that FuseBot can leverage deformable RF kernels to more accurately model deformable RFID tagged objects. The aim of this benchmark is to evaluate the performance improvement of this model. We evaluated FuseBot with both spherical and deformable RF kernels. We ran 20 trials across multiple scenarios where at least one RFID tagged item was deformable and FuseBot was tasked with retrieving a non-tagged target item that was fully occluded under the piles. In order to ensure a fair comparison, we did not include failed grasp attempts in the total number of actions for this microbenchmark as they were caused by grasping network errors rather than RF Kernels.

Table 4 compares the number of actions needed to retrieve target item when using deformable RF kernels compared to spherical RF kernels. We make the following remarks:

- FuseBot with deformable RF kernels retrieved the target object with median of 3.0 actions and 90th percentile of 4.0 actions. However, FuseBot with spherical kernel required a median of 4.0 actions and 90th percentile of 6.2 actions to finish the same tasks. This demonstrates that accounting for object deformability in RF kernels further improves the system's efficiency.
- Importantly, FuseBot with spherical kernels was still able to successfully retrieve the target object in all trials. This shows that despite decreased efficiency, FuseBot's probabilistic approach still allows for successful task completion despite inaccurate kernel models.

8.2.5 RFID localization accuracy

In our final microbenchmark, we evaluated the accuracy of FuseBot's RFID localization over 37 experiments. To evaluate the impact of occlusions on RFID localization accuracy, we computed the error in two cases: one where the tag was in line-of-sight (LOS) to the antennas and one where the tag was in non-line-of-sight (NLOS) (e.g., covered by clothes, stuffed animals, etc). We used the Optitrack motion capture system (Optitrack, 2017) to obtain accurate ground truth locations. Since the RF signal can emanate from any position on the RFID tag, we measure the error as the L2 norm between the estimated RFID location and the nearest point on the RFID tag.[10]

Table 5 shows the RFID localization accuracy in LOS, NLOS and overall. We make the following remarks:

- FuseBot is able to accurately localize RFIDs, achieving a median of 3.6 cm and a 90th percentile of 6 cm of error. We note that this level of error is typically less than the dimensions of the object to which the RFID object is attached, allowing FuseBot to accurately model the environment. We also note that FuseBot's probabilistic approach is specifically designed to account for these small errors.
- The localization accuracy in LOS and NLOS scenarios is very similar, with the median error increasing by less than half a cm and the 90th percentile increasing by 1 cm in NLOS scenarios. This is expected since RF signals can go through most occlusions, and this

[10] We performed a one-time calibration to remove offsets from the localization.

Table 5 RFID tags localization error

Localization environment	RFID localization accuracy (m)		
	10th pctl	Median	90th pctl
LOS	0.015	0.034	0.055
NLOS	0.023	0.038	0.065
Overall	0.017	0.036	0.060

The table shows the 10th, 50th, and 90th percentiles of L2 norm of localization error of RFIDs in line of sight, non line of sight, and all scenarios

matches results reported in state-of-the-art RFID localization work (Boroushaki et al., 2021b).

9 Discussion and conclusion

This paper presented FuseBot, the first RF-Visual mechanical search system that leverages RF perception to efficiently retrieve both RF-tagged and non-tagged items in the environment. The paper presents novel primitives for RF-Visual mapping and extraction and implements them into a real-time prototype evaluated in practical and challenging real-world scenarios. Our evaluation demonstrated that the mere existence of RFID-tagged items in the environment can deliver important efficiency gains to the mechanical search problem.

Our evaluation of FuseBot in end-to-end retrieval tasks also revealed a number of interesting insights. While FuseBot's design focused on retrieving untagged target items, our results showed that its efficiency in extracting RFID tagged target objects matches that of state-of-the-art RF-Visual mechanical search systems that can only extract RFID-tagged objects. Our evaluation also showed that FuseBot is successful and efficient in performing mechanical search across piles with deformable objects.

In conclusion, with the rapid and widespread adoption of RFID tags across various industries, this paper uncovers how RF perception can play a role in making robotic tasks more efficient and reliable for various industries such as warehousing, manufacturing, retail, and others.

Acknowledgements We thank the Signal Kinetics group for their help and feedback. This research is sponsored by an NSF CAREER Award (CNS-1844280), the Sloan Research Fellowship, and the MIT Media Lab.

Author Contributions TB, LD, and NN wrote the software code; TB, LD, and NN conducted experiments; TB, LD, and NN designed the figures; TB, LD, NN, and FA wrote the manuscript. TB, LD, NN, and FA conceived and conceptualized the method.

Funding 'Open Access funding provided by the MIT Libraries' This research is sponsored by an NSF CAREER Award (CNS-1844280), the Sloan Research Fellowship, and the MIT Media Lab.

Declarations

Conflict of interest F.A. is founder of Cartesian Systems. The remaining authors declare no competing interests.

Ethical statement The authors declare that this manuscript is an extended version of a conference paper presented at RSS 2022, titled "FuseBot: RF-Visual Mechanical Search." This manuscript includes new techniques (Sect. 4.5.3) and additional real-world performance evaluations in Sect. 8.2.4 and Sect. 8.2.5 with 57 new real-world trials, which have not been previously published or submitted elsewhere. The authors affirm that the research reported in this manuscript is original, and that the data and findings presented are accurate and reliable. The authors have disclosed any potential conflicts of interest, financial or otherwise.

References

Avigal, Y., Satish, V., Tam, Z., Huang, H., Zhang, H., Danielczuk, M., Ichnowski, J., & Goldberg, K. (2021). AVPLUG: Approach vector planning for unicontact grasping amid clutter. In *2021 IEEE 17th international conference on automation science and engineering (CASE)* (pp. 1140–1147). IEEE.

Aydemir, A., Sjöö, K., Folkesson, J., Pronobis, A., & Jensfelt, P. (2011). Search in the real world: Active visual object search based on spatial relations. In *2011 IEEE international conference on robotics and automation* (pp. 2818–2824). IEEE.

Bajcsy, R. (1988). Active perception. *Proceedings of the IEEE, 76*(8), 966–1005.

Bohg, J., Hausman, K., Sankaran, B., Brock, O., Kragic, D., Schaal, S., & Sukhatme, G. S. (2017). Interactive perception: Leveraging action in perception and perception in action. *IEEE Transactions on Robotics, 33*(6), 1273–1291.

Boroushaki, T., Leng, J., Clester, I., Rodriguez, A., & Adib, F. (2021a). Robotic grasping of fully-occluded objects using rf perception. In *2021 international conference on robotics and automation (ICRA)*. IEEE.

Boroushaki, T., Perper, I., Nachin, M., Rodriguez, A., & Adib, F. (2021b). RFusion: Robotic grasping via RF-visual sensing and learning. In *Proceedings of the 19th ACM conference on embedded networked sensor systems*. SenSys '21 (pp. 192–205). Association for Computing Machinery.

Chen, Y., Zhang, Z., Cao, Y., Wang, L., Lin, S., & Hu, H. (2020). RepPoints V2: Verification meets regression for object detection. *Advances in Neural Information Processing Systems, 33*, 5621–5631.

Curlander, J. C., & McDonough, R. N. (1991). Synthetic aperture radar (Vol. 11).

Danielczuk, M., Angelova, A., Vanhoucke, V., & Goldberg, K. (2020). X-ray: Mechanical search for an occluded object by minimizing support of learned occupancy distributions. In *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 9577–9584). IEEE.

Danielczuk, M., Angelova, A., Vanhoucke, V., & Goldberg, K. (2020). X-ray: Mechanical search for an occluded object by minimizing support of learned occupancy distributions. In *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 9577–9584). IEEE.

Danielczuk, M., Angelova, A., Vanhoucke, V., & Goldberg, K. (2021). X-ray code. https://github.com/BerkeleyAutomation/xray

Danielczuk, M., Kurenkov, A., Balakrishna, A., Matl, M., Wang, D., Martín-Martín, R., Garg, A., Savarese, S., & Goldberg, K. (2019). Mechanical search: Multi-step retrieval of a target object occluded by clutter. In *2019 international conference on robotics and automation (ICRA)* (pp. 1614–1621). IEEE.

Danielczuk, M., Matl, M., Gupta, S., Li, A., Lee, A., Mahler, J., & Goldberg, K. (2019). Segmenting unknown 3D objects from real depth images using mask R-CNN trained on synthetic data. In *2019 international conference on robotics and automation (ICRA)* (pp. 7283–7290). IEEE.

Dogar, M., Hsiao, K., Ciocarlie, M., & Srinivasa, S. (2012). Physics-based grasp planning through clutter. In *Proceedings of robotics: Science and systems (RSS '12)*.

EPC UHF Gen2 air interface protocol. http://www.gs1.org/epcrfid/epc-rfid-uhf-air-interface-protocol/2-0-1

Ettus Research, CDA-2990. https://moveit.ros.org/

Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., & Burgard, W. (2013). OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*.

Huang, H., Dominguez-Kuhne, M., Satish, V., Danielczuk, M., Sanders, K., Ichnowski, J., Lee, A., Angelova, A., Vanhoucke, V., & Goldberg, K. (2020). Mechanical search on shelves using lateral access X-ray. In *2021 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 2045–2052). IEEE.

Huang, X., Walker, I., & Birchfield, S. (2012). Occlusion-aware reconstruction and manipulation of 3D articulated objects. In *2012 IEEE international conference on robotics and automation* (pp. 1365–1371). IEEE.

Inlay, S. S. (2021). www.smartrac-group.com

Intel RealSense. (2019). https://www.intelrealsense.com

Kent Electronics. (2021). http://www.wa5vjb.com

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Advances in neural information processing systems*.

Liu, S., & Deng, W. (2015). Very deep convolutional neural network based image classification using small training sample size. In *2015 3rd IAPR Asian conference on pattern recognition (ACPR)* (pp. 730–734).

Luo, Z., Zhang, Q., Ma, Y., Singh, M., & Adib, F. (2019). 3D backscatter localization for fine-grained robotics. In *16th USENIX symposium on networked systems design and implementation (NSDI 19)* (pp. 765–782).

Ma, Y., Selby, N., & Adib, F. (2017). Minding the billions: Ultra-wideband localization for deployed RFID tags. In *Proceedings of the 23rd annual international conference on mobile computing and networking (MobiCom)* (pp. 248–260).

Morrison, D., Corke, P., & Leitner, J. (2018a). Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach. In *Robotics: Science and systems XIV (RSS)*.

Morrison, D., Corke, P., & Leitner, J. (2018b). GG-CNN code. https://github.com/dougsm/ggcnn

Nuand. (2021). BladeRF 2.0 Micro. https://www.nuand.com/bladerf-2-0-micro/

Optitrack. (2017). http://www.optitrack.com.

Price, A., Jin, L., & Berenson, D. (2019). Inferring occluded geometry improves performance when retrieving an object from dense clutter. arXiv:1907.08770

Robotiq. (2019). https://robotiq.com/products/2f85-140-adaptive-robot-gripper

Tse, D., & Viswanath, P. (2005). Fundamentals of wireless communication.

Universal Robots ROS Driver. (2020). https://github.com/UniversalRobots/Universal_Robots_ROS_Driver

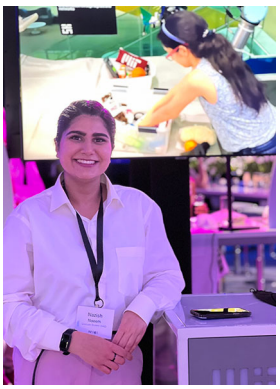
Universal Robots. (2021). UR5e. https://www.universal-robots.com/products/ur5-robot/

Wang, J., & Katabi, D. (2013). Dude, where's my card? RFID positioning that works with multipath and non-line of sight. In *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM* (pp. 51–62).

Wang, J., Adib, F., Knepper, R., Katabi, D., & Rus, D. (2013). RF-compass: Robot object manipulation using RFIDs. In *Proceedings of the 19th annual international conference on mobile computing & networking (MobiCom)* (pp. 3–14).

Tara Boroushaki is a Ph.D. student at MIT and a Microsoft Research Ph.D. fellow (2022). Her research focuses on fusing vision with radio frequency (RF) sensing through artificial intelligence. She designs algorithms and builds systems that leverage such fusion to enable capabilities that were not feasible before in applications spanning augmented reality, virtual reality, robotics, smart homes, and smart manufacturing. Her research on robotic perception and manipulation was named as one of the 103 ways MIT is making a better world (Finder of Lost Things). Her research has also been featured in public media including the Wall Street Journal, BBC, and World Economic Forum.

Laura Dodds is a Ph.D. student at MIT, where she works on wireless sensing. Her current research aims to develop novel RF sensing modalities that unlock new capabilities spanning applications in robotics, augmented reality, and supply chain. She was named an MIT Irwin Mark Jacobs and Joan Klein Jacobs Presidential Fellow in 2022 and her M.Eng. Thesis received the 2023 Charles & Jennifer Johnson MEng in AID Thesis Award. She received her M.Eng. and B.S. from MIT in 2022 and 2021.



Nazish Naeem is a Ph.D. student at MIT. Her research focuses on developing novel wireless sensing technologies for applications ranging from robotic manipulation to subsea Ocean IoT. She received her M.S. from MIT in 2023 and B.S. from LUMS in 2021.

Fadel Adib is an Associate Professor in the MIT Media Lab and the Department of Electrical Engineering and Computer Science. He is the founding director of the Signal Kinetics group, which invents wireless and sensor technologies for networking, health monitoring, robotics, and ocean IoT. He is also the founder & CEO of Cartesian Systems, a spinoff from his lab that focuses on mapping the physical world at unprecedented scale. Adib was named by Technology Review as one of the world's top 35 innovators under 35 and by Forbes as 30 under 30. His research on wireless sensing (X-Ray Vision) was recognized as one of the 50 ways MIT has transformed Computer Science, and his work on robotic perception (Finder of Lost Things) was named as one of the 103 Ways MIT is Making a Better World. Adib's commercialized technologies have been used to monitor thousands of patients with Alzheimer's, Parkinson's, and COVID-19, and he has had the honor to present his work to multiple heads of state, including President Obama at the White House. Adib is also the recipient of various awards including the NSF CAREER Award (2019), the ONR Young Investigator Award (2019), the ONR Early Career Grant (2020), the Google Faculty Research Award (2017), the Sloan Research Fellowship (2021), and the ACM SIGMOBILE Rockstar Award (2022), and his papers have won awards for best papers, demos, and highlights at premier academic venues including ACM SIGCOMM, ACM MobiCom, ACM CHI, IEEE RFID, Nature Electronics, and Nature Communications. Adib received his Bachelor's from the American University of Beirut (2011) and his Ph.D. from MIT (2016), where his thesis won the Sprowls Award for Best Doctoral Dissertation at MIT and the ACM SIGMOBILE Doctoral Dissertation Award.