Robotics: Science and Systems 2022
New York City, NY, USA, June 27-July 1, 2022

FuseBot: RF-Visual Mechanical Search

Tara Boroushaki, Laura Dodds, Nazish Naeem, Fadel Adib
Massachusetts Institute of Technology
{tarab, ldodds, nazishn, fadel}@mit.edu

***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 180 real-world experimental trials to evaluate FuseBot and compare its performance to a state-of-the-art vision-based system named X-Ray [10]. 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.

I. 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 [8, 10, 5, 16, 6]. 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 [8, 10, 16]. 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 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 [5, 6, 33]. 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.

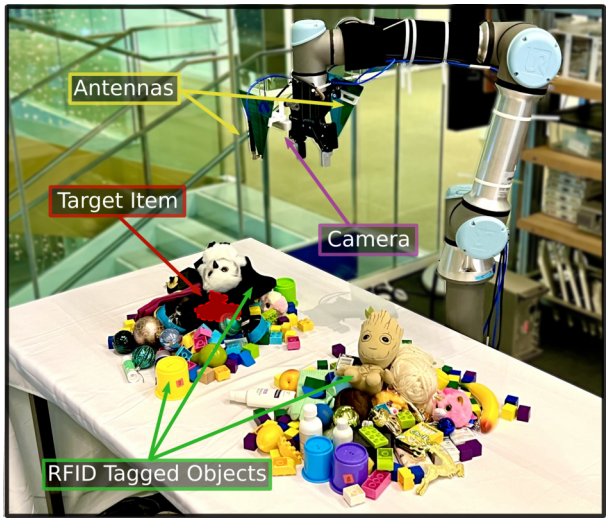


Fig. 1: **RF-Visual Mechanical Search.** FuseBot uses RF and visual sensor data (from wrist-mounted camera and antenna) to perform mechanical search and extract the occluded target items from the piles of both RFID tagged and non-tagged items.

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

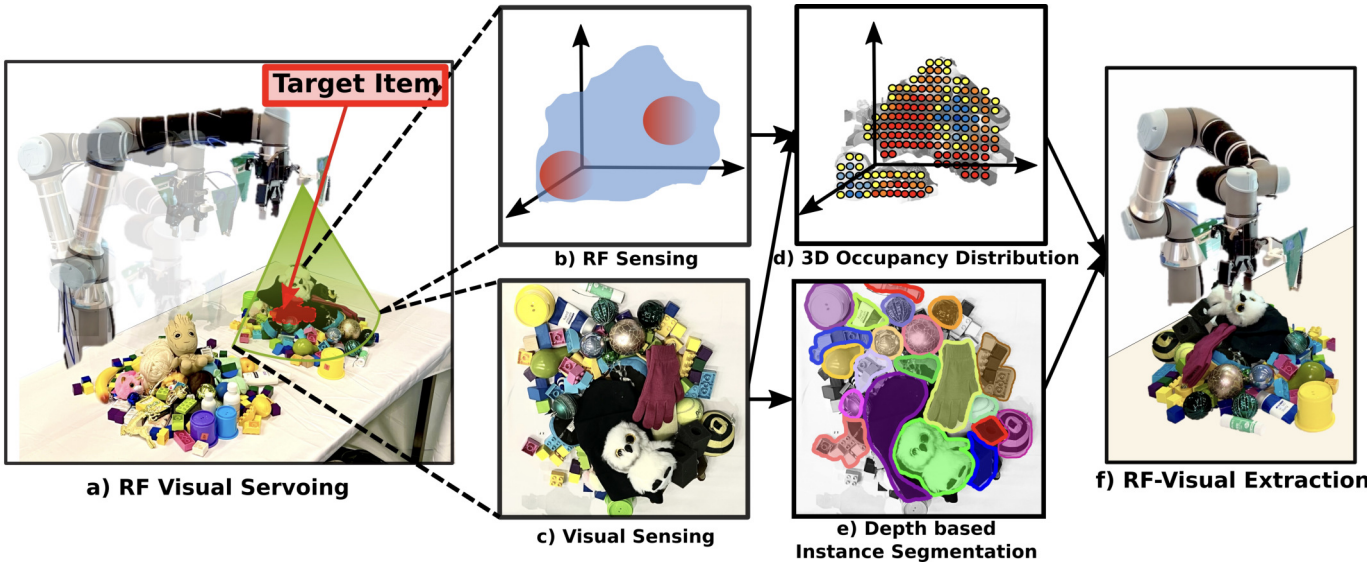


Fig. 2: **RF-Visual Mapping and RF-Visual Extraction.** (a) As FuseBot moves, it observes the environment using the wrist mounted camera and RF module. (b) Using the RF measurements, FuseBot localizes the RFID tagged items in the environment and computes RF kernels. (c) Using the wrist mounted camera, 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.

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 Figure 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.

[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.

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. 2(a). 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. 2(b). By combining this information with its visual observation of the 3D pile geometry (shown Fig. 2(c)), as well as prior knowledge of the target object's geometry, FuseBot creates a 3D occupancy distribution, shown as a heatmap in Fig. 2(d), 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 IV 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. 2(e). 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 section V.

We implemented a real-time end-to-end prototype of FuseBot with a Universal Robot UR5e [31] and Robotiq 2f-85 gripper [29]. As shown in Figure 1, we mount an Intel RealSense Depth camera D415 [19] 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 [27].

We ran over 180 real-world experimental trials to evaluate FuseBot. We compared our system to a state-of-the-art system called X-Ray [10], 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 90^{th} 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 $2.5\times$ improvement over X-Ray in environments where 25% of (non-target) items are RF-tagged and $4\times$ 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.

II. 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 [17, 28], 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 [2, 3, 4]. Other areas of research focused on efficiently grasping partially occluded objects using physics-based planners [13]. 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 [8]. 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 [10]. Other work has also looked at lateral search, where objects are retrieved from the side rather than from a pile [16, 1]. 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 section VII) 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 [5, 6, 33]. 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 [24, 32, 23]. 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.

III. 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.

IV. 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.

A. 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 [6].

FuseBot combines the visual information obtained during its trajectory using an Octomap structure [15]. 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:

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

[2]In our implementation, each voxel is a $2.5 \times 2.5 \times 2.5$cm cubic volume. 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. 3(a). Here, we can see that the regions under the surface of the piles have a high probability (red) of containing the target object.

B. RF Certainty Map

So far, we have explained how FuseBot constructs a 3D uncertainty map based on the camera's depth information. Next, we explain how it constructs a certainty map based on RF measurements.

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. We explain this process at a high level and refer the reader to prior work on RFID localization for more detail[24, 23, 6, 5]. 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 leverages each tag's response to compute the distance to the tag. As the robot moves above the pile to collect different depth measurements (as discussed in section IV-A), it can simultaneously collect distance measurements from each of the tags, then combine these measurements via trilateration to localize each of the RFIDs in the pile.

FuseBot leverages 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:

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

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

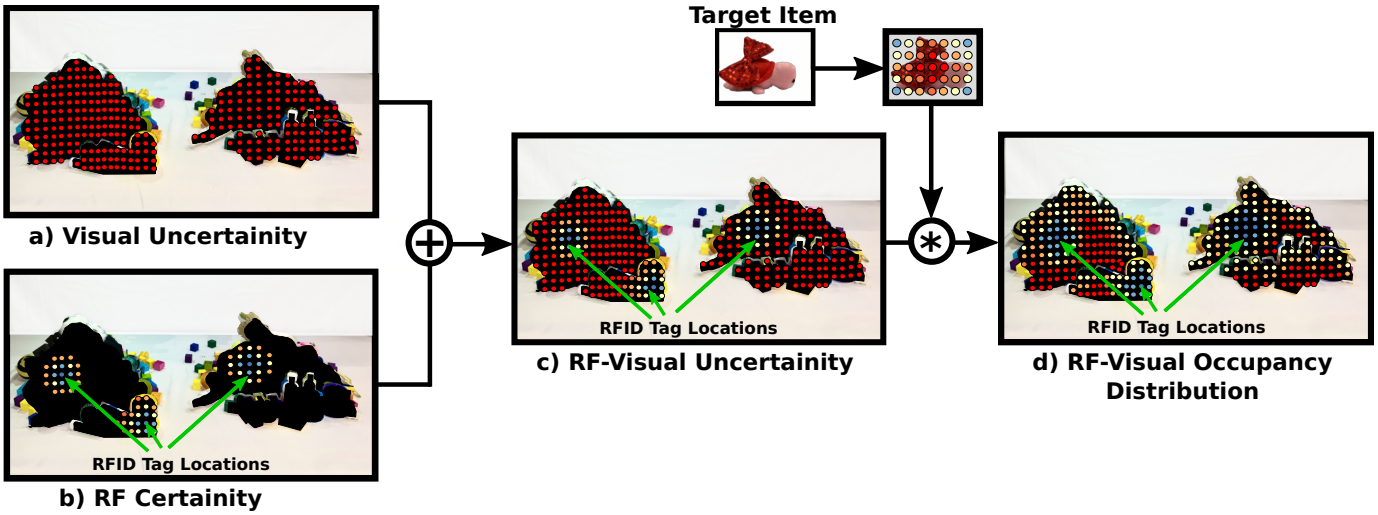


Fig. 3: **RF-Visual Mapping.** FuseBot a) constructs an initial map of unknown regions using visual RGB-D information and b) uses RFID tag locations to construct RF kernels. c) It then combines the RF and Visual information to more accurately map probable target locations. d) Finally, it uses the target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

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 i$^{\text{th}}$ RFID location, and $m(p, p_i)$ is the i$^{\text{th}}$ RF kernel. The RF certainty distribution for the example scenario (described in Fig.1) is shown in Fig. 3(b). 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. 3(c) 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.

C. 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 section IV-B. Specifically, it creates a target occupancy kernel that summarizes all the possible orientations of a target object using the following target gaussian kernel:

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

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.[3]

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.

Fig. 3(d) 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.

Generalizing to other scenarios: Our discussion in this section 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:

- When the target object is RFID tagged and not in the line of sight, FuseBot uses the calculated RF kernel in order

[3]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).

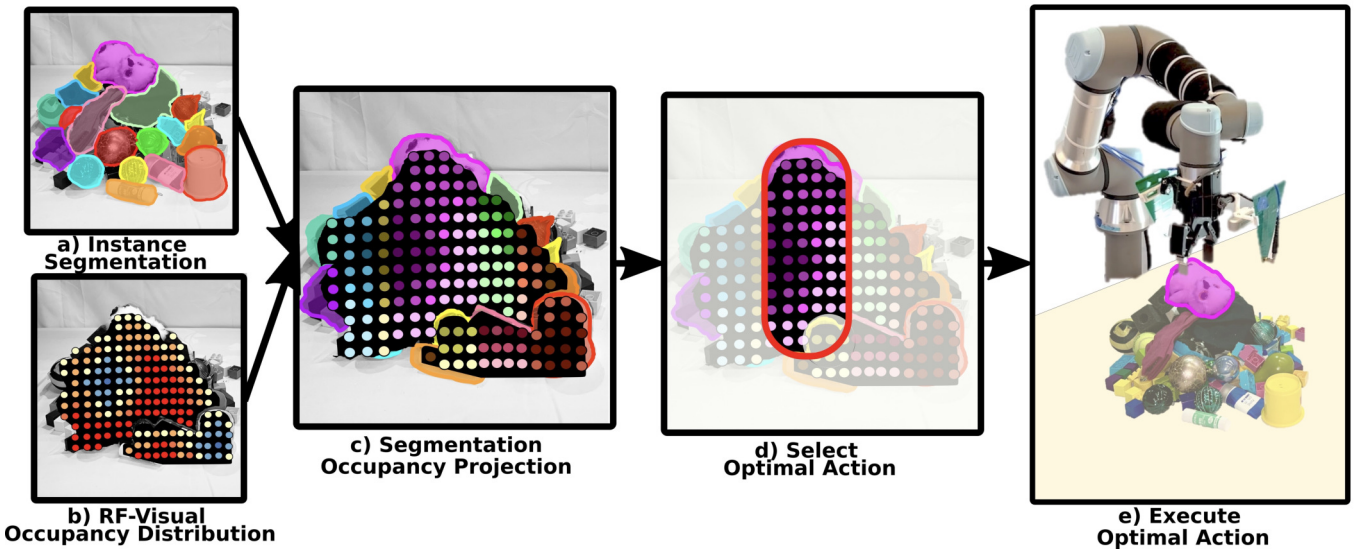


Fig. 4: **RF-Visual Extraction.** a) FuseBot performs depth based object segmentation to separate different objects in the environment. b) FuseBot uses the 3D occupancy distribution of the target item. c) FuseBot projects the occupancy distribution on each segmented mask. d) FuseBot sums the projected distribution on the area of each mask, and then chooses the mask with the highest sum. d) FuseBot chooses the next-best-grasp to extract the target item.

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.

- 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 [7, 8, 21, 22].
- Finally, it is worth noting that FuseBot's approach extends to deformable objects. In particular, even though the kernels (RF kernel and target kernel) leverage an object's bounding box, they only use this information to decrease the likelihood of certain regions, but do not eliminate them completely. The success in working with deformable objects is demonstrated empirically in VIII.

V. 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. 4(a). Next, in Fig. 4(c), 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:

$$s_i = \sum_{x,y \in m_i} \sum_{z=0}^{z_{m_i}} \gamma^{\frac{(z_{m_i} - z)}{0.025}} \times p_{x,y,z} \quad (1)$$

where s_i is the score of mask i, m_i is all (x,y) points contained within the i[th] 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[4].

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:

$$g_i \leftarrow \max_{(x,y) \in m_i} g(x,y) \quad (2)$$

where g_i is the best grasp probability for the i[th] 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 i[th] 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 i[th] 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[5]. 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 $\times$ 10cm $\times$ 10cm region around the grasp point, as well as determine if the target object was uncovered by the latest grasp.
- At any point, if FuseBot identifies the target object, it ends the RF-Visual extraction policy and proceeds to grasping the target object.

[4]In our implementation, γ is set to 0.95

[5]In our implementation, τ is set to 0.8

Algorithm 1 RF-Visual Extraction Policy

while Grasp Actions $\leq$ 15 **do**
  SEGMENTATION
  Compute object segmentation with SDMRCNN[9]

  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

VI. Implementation

Physical Setup. We implemented FuseBot on a Universal Robots UR5e robot [31] with a Robotiq 2F-85 gripper [29]. We mounted an Intel Realsense D415 depth camera [19] and two WA5VJB Log Periodic PCB antennas (850-6500 MHz) [20] on the gripper. The antennas are connected to two Nuand BladeRF 2.0 Micro software radios [27] 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 [24, 6]. We used standard off-the-shelf UHF RFID tags (the Smartrac DogBone RFID [18]) that costs around 3-5 cents.

Control Software. The system was developed and tested on Ubuntu 20.04 and ROS Noetic. We used MoveIt [14] as the inverse-kinematic solver to control the robot through the UR Robot Driver package [30]. The visual map of the environment is created using Octomap [15]. We used Synthetic Depth (SD) Mask R-CNN [9] 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 [25, 26]. The baseline, X-Ray [10] was implemented based on the published code [12].

VII. Evaluation

A. 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. 5. The scenarios had between 1 and 3 distinct piles of items, 0-10

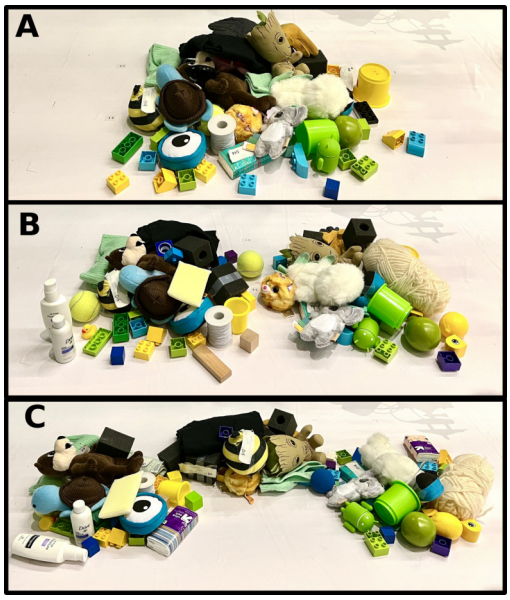


Fig. 5: **Example Evaluation Scenarios.** This shows some of the evaluation scenarios for A) 1 pile B) 2 piles, and C) 3 piles. The target item is fully occluded in all the scenarios.

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 FuseBot in scenarios where the target object was tagged with an RFID.

Similar to prior work [10] 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 [8] 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.

B. Baselines

We compared FuseBot's performance with X-Ray [11]. 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.

C. 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 & 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.

VIII. Results

A. 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 Section VII(c) that an experimental trial is considered successful if the robot can find and retrieve the target item within 15 actions.

	Number of Actions			
System	10^{th} pctl	Median	90^{th} pctl	Success Rate
FuseBot (Untagged)	2	3	6	95%
FuseBot (Tagged)	2	2	5	95%
X-Ray	2	5	11	84%

TABLE I: **Efficiency and Success Rate.** The table shows the success rate as well as the 10^{th}, 50^{th}, and 90^{th} 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.

1) Overall Number of Actions: Table I shows the 10^{th}, 50^{th}, and 90^{th} 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 90^{th} percentile of FuseBot with non-tagged items is 6 actions, while X-Ray's 90^{th} 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 90^{th} percentile.
- When searching for a tagged target item, FuseBot requires only 2 actions on median, and 5 actions for the 90^{th} 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 [6];[6] moreover, unlike these prior systems, FuseBot's benefits also extend to non-tagged items.

2) End-to-end Success Rate: Table I 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.

3) Search & Retrieval Time: Table II 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 seconds at the median, while X-Ray's median is 142 seconds, showing more than 2x improvement over the baseline's performance.
- The 90$^{\text{th}}$ percentile of FuseBot is 132 seconds, while X-Ray requires a 90$^{\text{th}}$ percentile of 237 seconds, 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.

	Search & Retrieval Time (sec)		
System	10$^{\text{th}}$ percentile	Median	90$^{\text{th}}$ percentile
FuseBot (Untagged)	40	62	132
X-Ray	50	142	237

TABLE II: **Search & Retrieval Time.** The table shows the 10$^{\text{th}}$, 50$^{\text{th}}$, and 90$^{\text{th}}$ percentiles for the search and retrieval time of both FuseBot and X-Ray.

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.

Fig. 6a 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 10$^{\text{th}}$ and 90$^{\text{th}}$ percentiles. We make the following remarks:

- Across all levels of complexity, FuseBot outperforms the baseline in terms of both its median and 90^{th} 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.

B. Microbenchmarks

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

1) Number of RFID Tagged Items: Recall from IV-B 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.

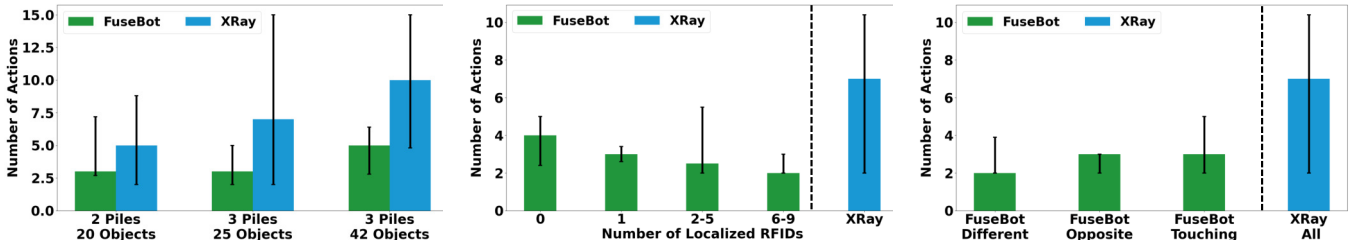
Fig 6b plots the number of actions required to retrieve the target item vs. the number of localized RFIDs in the environment for FuseBot (green) and X-Ray (blue). The error bars denote the 10$^{\text{th}}$ and 90$^{\text{th}}$ percentiles. Since X-Ray does not utilize RFIDs, the results are not separated 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.

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.

[6]See Fig. 14 in [6].



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

Fig. 6: **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 vs. 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 10^{th} and 90^{th} percentile respectively.

- *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.

Fig. 6c 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 10^{th} and 90^{th} 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 90^{th} 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 90^{th} is only 5 actions.

	Number of Actions		
Extraction Policy	10^{th} pctl	Median	90^{th} pctl
RF-Visual Extraction	2	2.5	4
Naive Extraction Policy	2.1	4	6.9

TABLE III: **Impact of Extraction Policy on Efficiency.** The table shows the 10^{th}, 50^{th}, and 90^{th} percentiles of the number of actions of FuseBot with different extraction policies

3) Impact of Extraction Policy: Next, in order to evaluate the benefits of FuseBot's RF-Visual extraction policy, we implemented a simpler extraction policy that does not optimize for information gain. The simpler policy operates in two steps: first, it 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 5cm of the voxel's projection on the surface of the pile.

Table III shows the 10^{th}, 50^{th}, and 90^{th} 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 90^{th} 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.

IX. Discussion & 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. As the research evolves, it would be interesting to explore how incorporating more complex models that account for deformability would allow FuseBot to achieve even higher efficiencies.

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.

ACKNOWLEDGMENTS

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

REFERENCES

[1] Yahav Avigal, Vishal Satish, Zachary Tam, Huang Huang, Harry Zhang, Michael Danielczuk, Jeffrey Ichnowski, and Ken Goldberg. Avplug: Approach vector planning for unicontact grasping amid clutter. In *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, pages 1140–1147. IEEE, 2021.

[2] Alper Aydemir, Kristoffer Sjöö, John Folkesson, Andrzej Pronobis, and Patric Jensfelt. Search in the real world: Active visual object search based on spatial relations. In *2011 IEEE International Conference on Robotics and Automation*, pages 2818–2824. IEEE, 2011.

[3] Ruzena Bajcsy. Active perception. *Proceedings of the IEEE*, 76(8):966–1005, 1988.

[4] Jeannette Bohg, Karol Hausman, Bharath Sankaran, Oliver Brock, Danica Kragic, Stefan Schaal, and Gaurav S Sukhatme. Interactive perception: Leveraging action in perception and perception in action. *IEEE Transactions on Robotics*, 33(6):1273–1291, 2017.

[5] Tara Boroushaki, Junshan Leng, Ian Clester, Alberto Rodriguez, and Fadel Adib. Robotic grasping of fully-occluded objects using rf perception. In *2021 International Conference on Robotics and Automation (ICRA)*. IEEE, 2021.

[6] Tara Boroushaki, Isaac Perper, Mergen Nachin, Alberto Rodriguez, and Fadel Adib. Rfusion: Robotic grasping via rf-visual sensing and learning. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, SenSys '21, page 192–205, New York, NY, USA, 2021. Association for Computing Machinery.

[7] Yihong Chen, Zheng Zhang, Yue Cao, Liwei Wang, Stephen Lin, and Han Hu. Reppoints v2: Verification meets regression for object detection. *Advances in Neural Information Processing Systems*, 33, 2020.

[8] Michael Danielczuk, Andrey Kurenkov, Ashwin Balakrishna, Matthew Matl, David Wang, Roberto Martín-Martín, Animesh Garg, Silvio Savarese, and Ken Goldberg. Mechanical search: Multi-step retrieval of a target object occluded by clutter. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 1614–1621. IEEE, 2019.

[9] Michael Danielczuk, Matthew Matl, Saurabh Gupta, Andrew Li, Andrew Lee, Jeffrey Mahler, and Ken Goldberg. 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)*, pages 7283–7290. IEEE, 2019.

[10] Michael Danielczuk, Anelia Angelova, Vincent Vanhoucke, and Ken Goldberg. 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)*, pages 9577–9584. IEEE, 2020.

[11] Michael Danielczuk, Anelia Angelova, Vincent Vanhoucke, and Ken Goldberg. 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)*, pages 9577–9584. IEEE, 2020.

[12] Michael Danielczuk, Anelia Angelova, Vincent Vanhoucke, and Ken Goldberg. X-ray code. https://github.com/BerkeleyAutomation/xray, 2021.

[13] Mehmet Dogar, Kaijen Hsiao, Matei Ciocarlie, and Siddhartha Srinivasa. Physics-based grasp planning through clutter. In *Proceedings of Robotics: Science and Systems (RSS '12)*, July 2012.

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

[15] Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 2013.

[16] Huang Huang, Marcus Dominguez-Kuhne, Vishal Satish, Michael Danielczuk, Kate Sanders, Jeffrey Ichnowski, Andrew Lee, Anelia Angelova, Vincent Vanhoucke, and Ken Goldberg. Mechanical search on shelves using lateral access x-ray. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2045–2052. IEEE, 2020.

[17] Xiaoxia Huang, Ian Walker, and Stan Birchfield. Occlusion-aware reconstruction and manipulation of 3d articulated objects. In *2012 IEEE International Conference on Robotics and Automation*, pages 1365–1371. IEEE, 2012.

[18] Smartrac Shortdipole Inlay. www.smartrac-group.com, 2021.

[19] Intel RealSense. https://www.intelrealsense.com, 2019.

[20] Kent Electronics. http://www.wa5vjb.com, 2021.

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[22] Shuying Liu and Weihong Deng. Very deep convolutional neural network based image classification using small training sample size. In *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, pages 730–734, 2015.

[23] Zhihong Luo, Qiping Zhang, Yunfei Ma, Manish Singh, and Fadel Adib. 3d backscatter localization for fine-grained robotics. In *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*, pages 765–782, 2019.

[24] Yunfei Ma, Nicholas Selby, and Fadel Adib. Minding the billions: Ultra-wideband localization for deployed rfid tags. In *Proceedings of the 23rd annual international conference on mobile computing and networking (MobiCom)*, pages 248–260, 2017.

[25] Douglas Morrison, Peter Corke, and Jurgen Leitner. Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach. In *Robotics: Science and Systems XIV (RSS)*, 2018.

[26] Morrison, Douglas and Corke, Peter and Leitner, Jürgen. Gg-cnn code. https://github.com/dougsm/ggcnn, 2018.

[27] Nuand, BladeRF 2.0 Micro. https://www.nuand.com/bladerf-2-0-micro/, 2021.

[28] Andrew Price, Linyi Jin, and Dmitry Berenson. Inferring occluded geometry improves performance when retrieving an object from dense clutter. *ArXiv*, abs/1907.08770, 2019.

[29] Robotiq. https://robotiq.com/products/2f85-140-adaptive-robot-gripper, 2019.

[30] Universal Robots ROS Driver. https://github.com/UniversalRobots/Universal_Robots_ROS_Driver, 2020.

[31] Universal Robots, UR5e. https://www.universal-robots.com/products/ur5-robot/, 2021.

[32] Jue Wang and Dina Katabi. 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*, pages 51–62, 2013.

[33] Jue Wang, Fadel Adib, Ross Knepper, Dina Katabi, and Daniela Rus. Rf-compass: Robot object manipulation using rfids. In *Proceedings of the 19th annual international conference on Mobile computing & networking (MobiCom)*, pages 3–14, 2013.