Wearable RFID Reader for Tracking Interactions with Ordinary Objects

Youssef Tobah and Rahul Gangwani

***Abstract*—RFID tag and reader technology has been well researched and developed for tracking user interactions with objects in environments that allow for stationary readers. Existing models consist of a large reader fixed to the ceiling or lights of a room, which requires an expensive reader, and cannot track any objects outside of this single reader's range. Thus, we propose a portable, wrist-worn RFID reader that can seamlessly track user interactions with tagged objects. The device combines an RFID reader, for identifying nearby objects, and a piezoelectric sensor for detecting object touches. We demonstrate the capability to detect and distinguish between multiple tagged objects that are close to the user's hand and recognize when the user has touched each object. Such a device provides the basis for tracking all interactions with tagged objects in a relatively low-cost, light-weight manner, free of the limitations imposed by a stationary reader.**

I. Introduction

With the increasing proliferation of embedded devices in everyday environments, there are numerous opportunities to leverage such systems for improving everyday life. One such improvement lies in allowing for users to track their interactions with common, everyday objects. For example, developers of virtual reality (VR) and augmented reality (AR) applications allow users to visually gain information about the objects around them.

However, there is currently no way to incorporate interactions with ordinary objects in VR and AR technology [11]. For VR games or for technicians using AR to obtain information about the machine they are working on, having the ability to track object interactions expands the capabilities of these applications. If technicians could track the interactions with and movements of all their tools, it would simplify the process of training newer technicians, as their movements could automatically be compared to and guided by previously tracked interactions of a veteran at work. Additionally, such tracking can make it easier to identify sources of error if a system fails, as all the interactions with the system leading up to the failure can be traced. VR games can also be improved by allowing interactions with real world objects to affect events in the virtual world.

One idea for implementing such a system is to equip objects with IMUs and sensors for tracing interactions. However, deploying such sensors to the hundreds of objects humans interact with daily would quickly become quite expensive and would make objects bulkier. The same issues apply to replacing everyday objects with smart objects that have such sensors built-in. Thus, for this project, we would like to explore the low-cost solution of applying inexpensive RFID tags to desired objects and using an RFID reader, worn on the user's wrist, to track each object's movements.

RFID is a long existing technology consisting of RFID chips, or tags, and an RFID reader. A reader can scan a chip or tag and identify nearby objects, as each tag has a unique ID. Recent research has successfully developed extremely low cost, passive RFID tags, which simply reflect signals emitted by a reader in order to communicate with that reader [14]. Since these tags are low cost, there is potential to widely deploy these tags, attaching them to hundreds of everyday, otherwise ordinary, objects, and allow an RFID reader to recognize and track interactions with such objects [14]. This fits neatly with the above motivation to immediately turn any object into a trackable smart-object.

However, this research uses a model that requires large-scale RFID readers to be placed in the ceiling or lights. This is both expensive and inconvenient for homes or buildings without easily removable ceiling panels. Additionally, if the ceiling is high relative to the floor, the reader will have poor accuracy for localization. Finally, such a system is limited to indoor use and cannot be easily extended to outdoor environments. Thus, to overcome the limitations of the current state-of-the-art [10], [14], [18], for this project, we present a low-cost, lightweight RFID reader that can be worn on one's wrist for detecting object interactions.

The reader will consist of an antenna that can detect nearby RFID tags and send any collected information to a microcontroller to identify the unique ID of the sensed tags. The first goal is to have the reader detect the presence of a nearby tagged object, essentially acting as an "RFID metal detector." At the same time, a passive vibration sensor is used to detect touch interaction with tagged objects to allow the system to identify what objects a user is touching. The RFID components will thus detect what object is closest to the user's hand, and the piezoelectric sensor will detect when the hand touches an object. If multiple objects are clustered together close to the user's hand, the system will directly use the vibration patterns to determine what object the user is touching.

In the case of direct classification for clustered objects, we observe that vibration patterns will vary between objects enough to classify a small number of different objects (2 or 3) based on inputs from the passive vibration sensor. Thus, we make the assumption that, if the objects are clustered together, the device will only be used for interaction with a small number of objects at a time. This still meets the needs of our use case, as a VR game can allow users to interact with a limited set of objects, still surpassing the capabilities of any currently existing VR technology, and technicians could use this device for their three most commonly used tools.

Future work can build on this project by using data from IMUs on board the microcontroller to track interactions with an object a user is touching. For example, the system could first detect a user is touching a tagged water bottle. The IMU data would then measure the movement of the water bottle while the user is touching it, and machine learning can be used to analyze the pattern of the IMU data and gauge whether the user has taken a drink or not. From here, the system could be extended to track use of a particular mechanical or medical tool, as mentioned in the above example use case. Therefore, the device developed in this project would form the basis of a low-cost wearable system that could track a user's interactions with tagged objects.

The rest of the paper is organized as follows. Section II gives background on RFID and vibration sensing used by the system. Section III explains the system design as a whole as well as the design and algorithms used by each module. Section IV describes the details of how each module was implemented, while Section V presents the results. The limitations of the prototype in its current form are discussed in Section VI. Finally, Section VII gives an overview related work on relevant topics and Section VIII concludes.

II. Background

The following section provides background on the two technologies that form the basis for this project: RFID and vibration sensing.

A. RFID

Radio Frequency Identification (RFID) is a form of wireless communication in which a transponder communicates with a reader through backscatter modulation [18]. The transponder consists of an integrated circuit (IC) containing a processing unit and memory unit as well as a printed antenna. Since the circuit allows for harvesting energy from RF signals, the transponder does not need a battery to source power, and can act as a passive tag.

In backscatter modulation, the reader sends out a 915 MHz carrier wave to all passive tags in the surrounding environment. The tags harness the energy from the received waveform in order to power the IC chip. When the tags have enough power, they modulate the carrier wave through a process called impedance matching [19]. With impedance matching, the tags can encode their ID by switching between two impedances: one that absorbs the carrier wave, and one that reflects. The reader can then decode the modulated waveform from each tag to receive the IDs of all surrounding objects.

The benefit of RFID communication comes from the passive tags. The tags act as "stickers" and can easily be placed on common household objects, as shown in Figure 1. Tags are also low in cost, size and weight [23]. However, the drawback of RFID comes from the readers, which are high-cost and consume much power. Frii's Equation, rearranged in terms of received power:

$$P_R = P_T - 20log(\frac{4\pi d}{\lambda}) + G_R + G_T - L_P \qquad (1)$$

Fig. 1: **RFID tag on an object.**

shows that the received power is proportional to the transmitted power and distance [20]. The wave transmitted from the reader is the same wave that is reflected from the tag. This means that in order for the reader to decode the modulated waveform, enough power must be transmitted to cover the distance to the tag and back. Readers are also integrated with an anti-collision protocol in order to decode multiple tags at the same time [10]. This means that readers must be robust and powerful enough to communicate with multiple tagged objects, which leads to overall higher cost.

B. Vibration Sensing

Piezoelectric sensors are sensors that induce an electric charge in the presence of an external force. These sensors have non-centrosymmetric crystals that respond to vibrations at a range of 1 Hz to several MHz [22]. In this way, piezoelectric sensors can be used to detect vibrations, either by touching the sensor or touching an object near the sensor, causing vibrations to propogate through the surface the sensor is resting on.

Vibsense [15] took advantage of vibration sensing to develop a system that identifies and localizes touch events based on the vibrations they cause. The paper is able to detect and identify objects via vibrations using the following steps:

1) Receive and filter vibration signals from high frequency noise
2) Convert signals to the frequency domain (FFT)
3) Obtain Power Spectral Density (quantity of power at each frequency)
4) Use PSD to classify presence of an object

However, as noted by the paper, since VibSense directly classifies objects based on their vibrations, it can only classify a limited number of objects. Furthermore, VibSense simply tells a user what objects are resting on a table by causing continuous vibrations on that table using a motor. It cannot be used to indicate what object a user has directly touched.

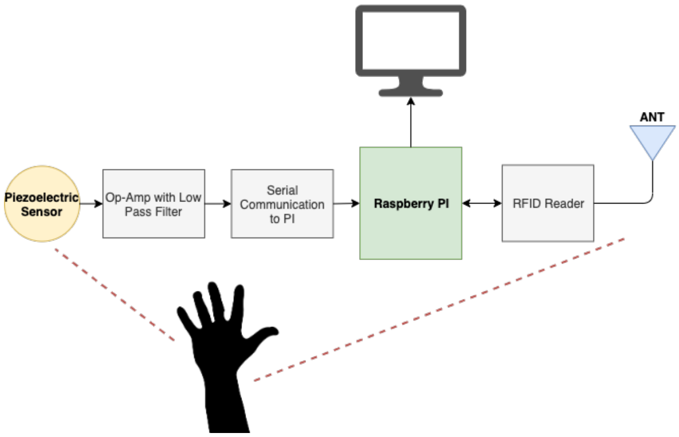

Fig. 2: **System block diagram.**

III. Design Overview

The proposed system combines RFID detection and vibration sensing to detect what object a user is touching. A diagram for the system is shown in Figure 2. On the right side of the diagram is the RFID antenna, which emits a signal to detect what objects are near the user's hand. RFID tags emit signals back to the antenna, and these signals are passed to a decoder which extracts the IDs of detected objects and sends them to a microcontroller. The piezoelectric sensor, shown on the left side, simultaneously gathers vibration data. The data is filtered and passed to the microcontroller for classifying whether the vibrations as object-touch or non-object-touch. Thus, RFID shows which objects are nearby and vibration sensing shows whether a user is touching an object; all together, the system can identify when a user touches and object and what object he or she is touching.

A. Identifying Nearby Objects

For this system, the role of RFID is to identify what object is closest to the user's hand. Localization via RFID is currently an open problem [13], [16], as past research attempted to track exact positions of tagged objects with results too inaccuate to be used for tracking the precise locations of individual objects. Additionally, such localization requires solving complex issues such as calculating the angle of arrival of a signal and overcoming phase wrapping. We therefore, instead devise a simpler method for detecting the nearest tagged object.

We begin by outputting a signal from the antenna at constant power. The radiation pattern of this signal depends on the nature of the antenna, and we assume this radiation is roughly the shape of a sphere. If multiple objects are detected, the signal's power is reduced, reducing the signal's range and the size of the sphere of radiation. The signal is reduced until only a single object is detected, showing which object is closest to the antenna. The device continues to reduce the power until it is at the minimum power possible such that the object is still within range. This minimum power level thus roughly shows how far the object is from the user's hand. If the object is within ”touching distance” the microcontroller is alerted and waits for a vibration signal to determine if the user is touching the nearby object. The touching distance threshold can be calibrated for each object upon first using the device.

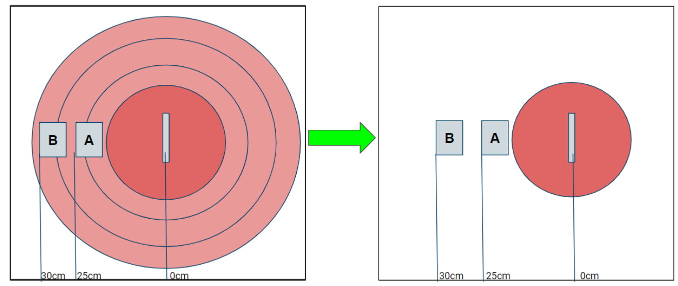

Fig. 3: **RF range changing to detect only a single object, but decreasing range by too much.**

Algorithm 1: RFID Range Variation

```
Set range to initial value;
Set RANGECHANGE to initial value;
if no objects detected then
    state = 0;
end
if single object detected then
    state = 1;
end
if more than one object detected then
    state = 2;
end
if state changed or state == 1 then
    RANGECHANGE = initial value;
else
    RANGECHANGE = RANGECHANGE * 2;
end
if state == 0 then
    range = range - RANGECHANGE;
end
if state == 2 then
    range = range + RANGECHANGE;
end
```

If no objects are within the antenna's range, the signal's power is amplified, expanding the size of the radiation sphere, until an object is detected. Thus if any tagged objects are within the antenna's maximum range, the device can show the user which object is closest to his or her hand.

The speed at which the antenna can lock-on to the closest objects depends on the amount the signal increases or decreases by as it adjusts toward having only the single closest object within its range. This can be modeled as a classical controls problem: if the signal changes by a large amount, it can more quickly approach the desired value, but is likely to overshoot beyond and acceptable range and would need to spend time readjusting. For example, visualized in Figure 3, if object A is 25cm away from the user's hand, object B is 30cm away, and the antenna outputs at a range of 40cm, both objects will be within range. The antenna needs to emit signals at smaller ranges until it detects only one object. If it reduces the range by 1cm each emission, reaching the desired value (to only have A within range) will be quite slow. However, if it reduces the range by 20cm, it will overshoot, emitting a signal where *neither* object is in range, requiring an adjustment period of increasing the range.

To find a balance that avoids constant overshooting but allows for quickly adjusting the range, we maintain states for object detection as shown in Algorithm 1. One state represents detecting no objects, a second state represents detecting a single object, and a third state detecting more than one object. If the signal's range increases or decreases and the object remains in the same state, the increase/decrease amount is doubled added to/subtracted from the signal once again. This process repeats until the state changes, which causes the increase/decrease amount to reset to its initial value.

B. Detecting Touch Events

While RFID tracks what objects are near a user's hand, the piezoelectric sensor is used to detect the moment the user touches a tracked object. When a user touches an object, the vibrations propagate through hand and are detected by a piezoelectric sensor. The analog voltages produced by the sensor are converted to digital values and stored by the microcontroller.

To differentiate between vibrations caused by noise and vibrations caused by touch events, the following techniques from VibSense [15] are used. First, the data is passed through a sliding window, and the average value of every point within the window is taken, using the equation

$$A(t) = \sum_{n=t}^{t+S} a^2(n), \tag{2}$$

where *t* is the start time of the window, *S* is the length of the window in samples, and *a(n)* is the amplitude of a sample recorded at time *n*. If this average amplitude passes a certain threshold, we save a fixed number of sample points starting at the beginning of the window and ending at a point that occurs a fixed number of time units later corresponding to the length of a touch event in seconds.

The recorded values are then converted to the frequency domain using the equation

$$PSD_i = 10 \log_{10} \frac{abs(FFT(r_i))^2}{f_s \times n}, \tag{3}$$

where n is the number of samples, r_i is the signal, f_s is the sampling rate, and *FFT()* represents a fast Fourier transform. With this we obtain the power spectral density (PSD) values of the touch event. This means the amplitude of every component frequency making up the recorded touch signal is known. These values are used as features for a machine learning classifier, where each frequency acts as a feature. Since it is expected that different types of touch events will produce signals with unique component frequencies, this data can be used for classification.

The classifier then uses the extracted features to determine whether the touch event corresponds to an object touch or a "false touch," which refers to vibration caused by any actions other than touching an object (e.g. touching the surface the object is resting on, touching one's own fingers, or shaking one's own hand). Therefore, a binary classifier can be used to classify between object-touches and false touches.

This presents a key difference from VibSense, which directly uses vibrations to classify objects. VibSense needs a class for every object that can be potetnially touched and differentiates classes based unique vibration patters produced by each object, limiting the number of objects it can potentially classify. Our proposed device, however, can determine *what* the object is using RFID and only uses vibration to determine *if* the user has touched an object. Using two broad classes (object and non-object), as opposed to many narrow classes (one class for each object) allows for identifying interactions with much larger set of objects than previous work.

Fig. 4: **The wearable device containing the required sensors.**

IV. IMPLEMENTATION

A prototype version of the system was designed in the form of a box shown in Figure 4 that can be strapped to the user's wrist. The box features an antenna for RFID sensing and a piezoelectric sensor that can be attached externally to the bottom of the box such that it makes contact with the user's skin. The sensors in the box were connected by wire to a Raspberry Pi. Future work can use components built into a smartwatch so that touch detection can be performed with hardware that is more readily available to the user.

To improve the accuracy for the proof of concept, we place the piezoelectric sensor on the back of the user's hand, as the closer the sensor is to the fingers, the stronger the vibration signals detected by the sensor.

A. RFID

The RFID modules consisted of a Taoglas PC.91 antenna for emitting and detecting RF signals, and a USBPro *ThingMagic* decoder module for extracting the tags' IDs from signals received by the antenna. A python wrapper for MercuryAPI [1] was used to control the decoder and antenna.

Single Object Detection To verify this hardware could accurately detect tags, we wrote a simple program that allowed the user to input the name of a desired object, and the system would show if the object was nearby. Observing that this could be useful for a metal-detector-like functionality of finding lost objects, we attached the wristband at the end of a stick and hid a tagged object out of sight. After inputting the name of the hidden object to the system, we waved the stick around our surroundings until the display showed the object was nearby, guiding us to the object's location. This demonstrates how the proposed device could easily be extended into an "object-detector" for finding lost items. Since the main focus of the project, however, was to detect touch interactions, we leave refining this object-search feature for future work.

Determining the Closest Object In addition to detecting whether a single given object is nearby, the RFID modules need to determine what object is closest to the user if multiple tagged objects lie within range. Thus, we implemented the range-variation algorithm described in Section III-A. As a preliminary test, we placed a tagged fork and tagged knife on a table and alternated between hovering one hand over the fork and over the knife while wearing the device. The algorithm was able to accurately control the antenna's range to tell whether the user's hand was closer to the fork or the knife.

B. Touch Detection

For the touch detection component, we ran the output of the piezoelectric sensor through an amplifying low-pass filter built with an OPA350 operation amplifier. A low-pass filter was used since, according to [15], vibrations produced by touching ordinary objects should have frequencies larger than 20kHz. We thus filter out all higher frequency signals to reduce noise, and the op-amp is used to further boost the signal to noise ratio.

Since the Raspberry Pi does not have an internal ADC, a MCP3008 ADC is used to convert the filtered analog vibration signal to digital values. The Pi communicates with the ADC using SPI, which is somewhat problematic since the maximum sampling rate is only 25kHz—less than the required 40kHz Nyquist sampling rate. However, due to the nature of the observed signals and the classification, the undersampling had little impact on the classifiers accuracy, which is discussed in more detail later within Section IV-B.

Feature Extraction As explained in Section III-B, the vibration data is passed through a sliding window, which takes the average of the signal's amplitude and considers the signal a touch event if this average passes a certain threshold. The length of the window is set to 0.2 seconds, corresponding to the estimated length of a touch event.

Time domain touch events are then converted to the frequency domain for feature extraction. The amplitudes of 2500 frequency components (over a range of 0 to 25kHz) are used to classify a touch event, where each frequency is a feature. Thus, the classifier is given 2500 features to use for its classification.

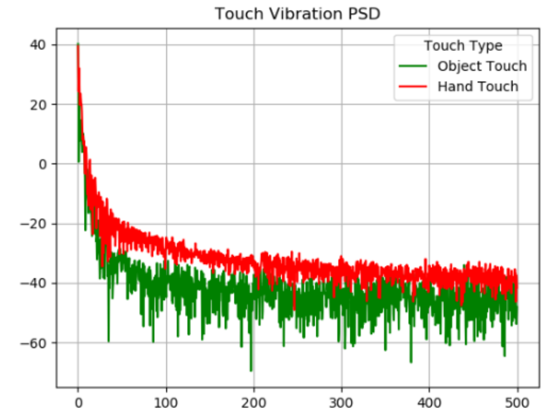

Fig. 5: **PSD values for an object touch and non-object touch.**

We observed example PSDs for an object touch and false touch and found noticeable differences in the amplitudes of each frequency component between the object touch and false touch. As shown in Figure 5, the large spikes in amplitude occur at different frequencies depending on whether the event is an object or false touch, showing that a classifier should be able to easily distinguish between the two.

Classification The machine learning algorithm is a binary classifier that distinguishes touching an object from non-object touches. To train the classifier, for each touch event, the PSD values are appended to an Excel spreadsheet. The last column of the spreadsheet is labeled with the touch event ("Object" or "Nothing"). When training, a DIP Switch is used to toggle between generating labels for object touches and non-object touches.

Scikit-learn, a machine learning library in Python [17] is used to generate a model to classify the touch event. The training set was run through multiple binary classification algorithms in order to understand which produced the most accurate model. Each model was validated through k-folds cross validation. In K-folds cross validation, the dataset is randomly split k times with a fraction of 1/k used as a test set [4]. Using Python, this was implemented in these steps:

1) Choose the number of folds, k, to be 10
2) Call train_test_split to split the dataset 90% train 10% test
3) Generate a model using the training set, and score the model by testing on the testing set
4) Append each score until k scores have been generated, and find the average to determine the accuracy of the model

After running these steps on multiple models, the algorithm with the highest accuracy was found to be Support Vector Machine (SVM) with Radial Basis Function (RBF) kernel. SVM is a supervised learning classifier which creates a hyperplane that divides data sets into two sides, with the dividing line representing the classifier's decision boundary. Based on the kernel function used, the hyperplane will take on a different shape to better distinguish the two classes [3]. The resulting model is incorporated on the Raspberry Pi to receive vibration data PSDs and provide classifications in real time.

The reason why SVM produced the best accuracy compared to the other models is that SVM works best for high dimensional data (when the number of features exceed the number of observations) [9]. In our case, we gather the values of 2500 different frequencies, which means 2500 features, and use roughly one tenth to one fifth as many observations. This is also shown in another study that found SVM with RBF kernel works best for vibration-based classifiers [6].

Undersampling. As previously mentioned, due to the Pi's limitations, we are forced to undersample vibration data, resulting in aliasing. However, since the aliased frequency component is dependent on the value of the true frequency component, it is still possible to classify based on aliased data. This will only become a problem if the aliased value happens to have the same frequency as the true frequency that is a defining feature for a different class. We found that even with this aliasing, frequencies produced by objects belonging to different classes rarely matched, as shown in Figure 5, causing little impact on accuracy.

V. Evaluation

A. Experimental Methodology

We first integrated the RFID modules (including the range-variation algorithm) with vibration sensing modules (including the binary classifier). Since the communicating with the RFID reader is relatively slow and could interfere with the vibration sensing sampling rate, the algorithms for the two sets of modules were executed in parallel using the Python thread library. We tested the system under the following four use cases, and the results are presented in the following section:

1) Detecting touches of a single object
2) Distinguishing touches of two objects at a set distance away from each other
3) Detecting touches of a single object on varying surfaces
4) Distinguishing clustered objects directly based on vibrations

Training. For each experiment, we trained the machine learning algorithm with 100 object touches for each potential object and an equal number of false touches. For example, when classifying between a cup and a bottle, the training data consisted of 100 cup touches, 100 bottle touches, and 200 false touches. The touch events of the first two sets are labeled as object touches and those of the third set as false touches.

Detecting touches a single object. In this experiment, a single tagged object was placed on a hard surface in front of the user. The user, with the wristband system and piezoelectric sensor attached, reached for and touched the object. The Raspberry Pi, connected to a monitor, displayed when the user's hand was near and object, when he touched the object, and showed the name of the object as well. The user also touched his own hand while close to the object to verify that the machine learning algorithm correctly classified the false touches. Figure 6 shows the output when the observer was interacting with the object (left), and when the observer was not touching the object (right).

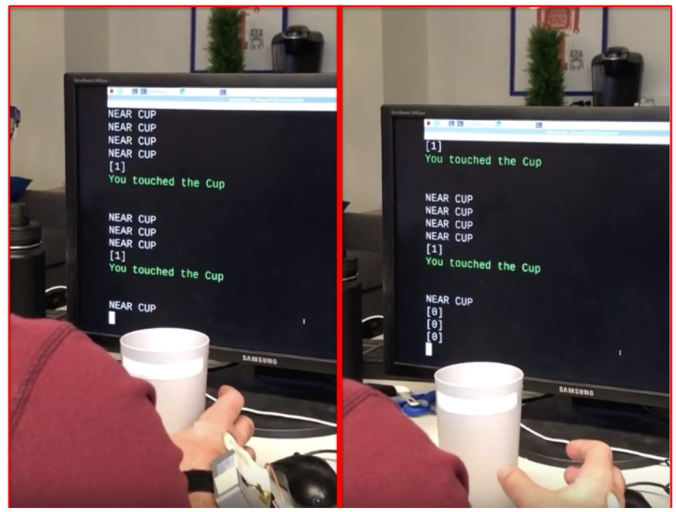

Fig. 6: **Single Object Touch:** [1] indicates touching object. [0] indicates touching nothing. NEAR CUP indicates RFID was in range and detecting only one object.

Distinguishing touches of two object at a distance. In this experiment, two tagged objects were placed on a hard surface approximately 18 inches from each other. The user placed his hand between the two objects, and alternated between touching one or the other. The user also touched his own hand and the surface the objects were resting on to verify correct classification. Figure 7 shows the experiment and the monitor display when the observer was touching a tagged hand sanitizer bottle (right), touching a tagged cup (left), and touching the surface near the cup (middle).

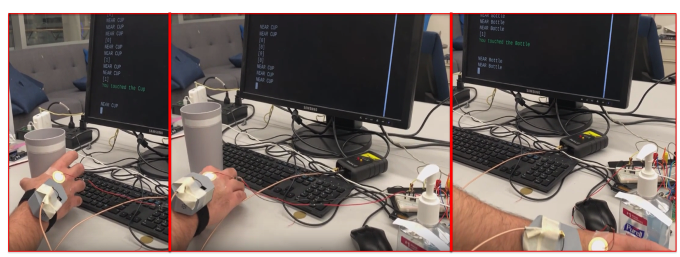

Fig. 7: **Two Object Touch:** [1] indicates touching object. [0] indicates touching nothing.

Detecting touches an object on various surfaces. In this experiment, a single tagged object rests on a hard surface and is touched by the user. The object is then moved to rest on a softer material in order to verify that the vibrations are not affected by the surface material to the point that it would interfere with the classifiers accuracy. Figure 8 shows the output correctly identifying a touch on a hard surface (left), and the output correctly identifying a touch on a softer surface (right).

Fig. 8: **Varying Surface Touch:** Softer surface used was a jacket (right).

Distinguishing two objects close together. One limitation of the range-variation algorithm is that if the objects are too close together, the range cannot decrease to a value which allows the antennta to identify only one object. However, according to VibSense, it's possible to identify objects based only on vibrations alone if the objects are made of different materials. In this experiment, two tagged objects of different material were placed close together, and the user alternated between touching one or the other. The binary classifier was modified in this experiment such that it classified between touching a cup or bottle rather than touching an object or not. Figure 9 shows the output when touching the bottle (left) or cup (right).

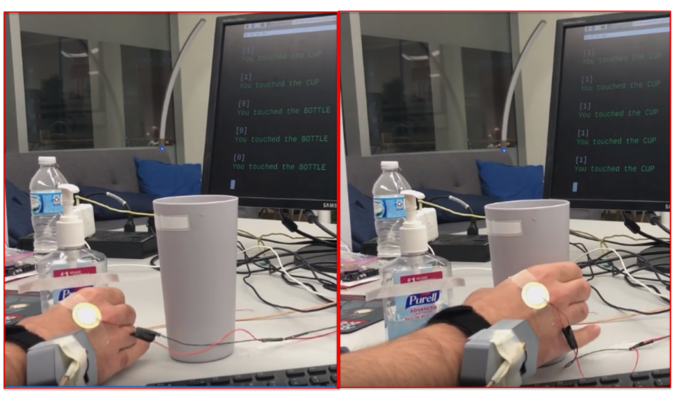

Fig. 9: **Touching Clustered Objects:** [1] indicates touching cup. [0] indicates touching hand sanitizer

B. Performance Results

Figure 10 shows the accuracies of each scenarios. For each scenario, two accuracies are reported: the first representing the accuracy from cross validation, determining the type of touch (blue), and the second representing accuracy from integrating RFID and vibration and attempting to determine both if and object is touched *and* what object is touched (red). The accuracies marked red were found by performing the experiments and determining the amount of times the correct output was given. The average experimental accuracy is approximately 90%.

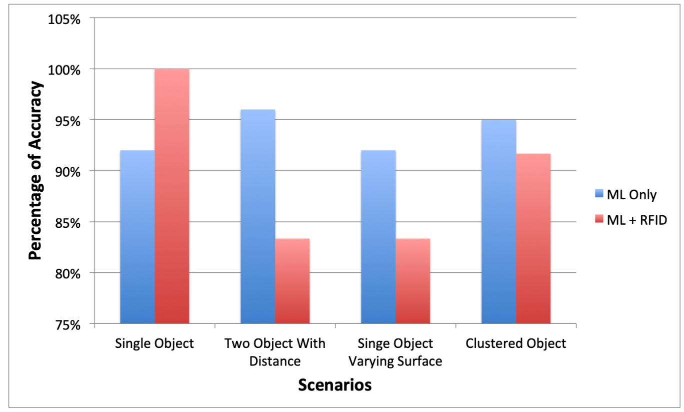

Fig. 10: **Accuracy of testing on different scenarios:** Blue columns show validation accuracy and red columns show experimental accuracy

Single object is the only case for which the experimental accuracy is higher than the validation accuracy. This is because, for the case of one object, RFID can very accurately tell if the object is close to the user's hand. Thus, if the machine learning classifier ever made a mistake, e.g. reporting an object touch if there was none, that faulty result would be thrown away since RFID could identify that the user's hand was too far away from the object to possibly be touching it.

As for the other cases, RFIDs accuracy decreases when multiple objects are within range. Now there are two possible sources of error, a machine learning misclassification, or RFID reporting that a farther object is closer to the user's hand. This results in less accuracy than the single-object case. Note that even though the machine learning alone provides higher accuracy, it is only because it is simply distinguishing between object-touches and non-object touches, while the combined accuracy has the more difficult task of distinguishing type touch *and* determining what object is being touched.

VI. Discussion

The system successfully met the initial goal of identifying interaction with nearby objects, and even surpassed this by distinguishing between multiple objects and including a metal detector feature. However, the prototype built for this project does have several limitations that we hope to improve for future work. The following sections discuss these limitations.

A. Machine Learning

Training While the machine learning algorithm was able to accurately detect when a user had touched an object, it required training a model each time the piezoelectric sensor was placed on a different location on the user's hand. Ideally, this system could be used with the sensor at any location on the user's hand with minimal calibration. Thus, future work can consider using a training set that consists of touches performed with the piezoelectric sensor placed at different points of the hand on different users.

Feature Extraction For feature extraction, we use 2500 equally spaced frequencies over a range of 25kHz that make up the frequency components of touch events. It is possible that some frequencies are more useful than others for classifying touch types, and thus future work can consider more careful ways of determining which frequencies are useful to reduce the number of frequencies used as features, decreasing noise and allowing for faster classification, while maintaining or improving accuracy.

B. Clustered Objects

Since the RFID algorithm assumes that a user will touch the object closest to his or her hand, the performance is poor when many objects are clustered together, which is why, for the case of clustered objects, we opted to classify directly based on an object's characteristic vibration, as shown in Section V-A. However, this has limitations similar to past work [15], as it requires a class for every object, limiting the amount of objects that can be practically classified. Future work can explore using alternate RFID algorithms, or rough localization of tagged objects, to provide more information on what object is closer to the user's fingers. This way, a classifier only needs to distinguish between object touches and false touches, expanding the number of objects that can be classified, as we have done for classifying non-clustered objects.

C. Many Objects

For a large number of objects, even if the objects were not tightly packed together, the RFID distance-varying algorithm has trouble determining which object is closest to the user's hand. The main cause of this issue is likely that the antenna's radiation pattern is not perfectly spherical, even though the algorithm assumes otherwise. Thus, future work can consider trying different antennas to find one with a radiation pattern that is more symmetrical, or that is radially polarized, meaning the radiation is focused in a beam pointing in a particular direction. This would provide much greater accuracy for locking on to the closest object.

D. IMU Tracking

A key point that would be desirable for future work would be to incorporate IMU data for detailed object interaction tracking. While the goal for the original project was to detect what object a user is touching, the ultimate goal is to have a device that fully understands how a user is interacting with tagged objects. To meet this need, light-weight accelerometers and gyroscopes could be equipped to the system to track a user's movement while he or she is holding an object. This information could then be passed to a machine learning algorithm to classify how a user is interacting with the objects, similar to the approach used in IDSense [14] but tracking user movement via IMUs versus tracking tagged object movement via RFID.

VII. Related work

Numerous other works propose methods of identifying objects and interactions with objects. EMSense [12] utilizes the fact that different objects emit different electromagnetic waves. This work implemented off line machine learning algorithms to classify objects based on the frequencies of emitted electromagnetic waves. However, this system was limited to work only with conductive devices, whereas our system works on any tagged object.

There have also been studies that have used acoustic sensing via microphones to detect vibrations and classify touch events [2], [8]. However, these devices required a microphone to be placed on the interacted object in order for the microphone to recognize a touch. Our system, on the other hand, uses passive vibrations, with a sensor to detect vibrations on the user's hand, in order to avoid adding external sensors to the target objects.

There has also been work focused on using either mounted or portable RFID readers alone to identify tagged objects [7], [14], [21]. However, these papers rely on touching or covering the tags themselves to identify touching the object, which limits the places that can be touched. With our solution that uses passive vibrations, it is possible to recognize touches that happen anywhere on the object.

Other works have considered using vibrations and piezoelectric sensors to classify touch events as well. As previously discussed, VibSense [15], places piezoelectric sensors on surfaces and causes constant vibrations to classify what objects rest on the surface. It cannot be used to tell what object a user is touching, however, and directly classifies objects based on their vibrations, reducing the amount of objects it can accurately classify. Additionally, Taprint [5], uses vibrations caused by a user tapping the back of his or her hand to detect key presses on a virtual keyboard projected from a smartwatch. It can accurately classify where on the hand the user touches, but cannot classify interactions beyond this, meaning it is not useful for identifying interactions with ordinary objects.

VIII. Conclusion

With increasing desire to enhance the capabilities of ordinary objects, much research has been dedicated to crafting new methods of tracking interactions with everyday objects. In this project, we have presented the first portable RFID reader that can detect touches of tagged objects, without requiring the user to touch the tags or use electromagnetic waves that are not present in most objects. We find that by combining RFID and vibration sensing, we can detect and identify object touches with 90% accuracy, and hope that this will act as the basis for a future device that can fully track interactions with tagged objects.

REFERENCES

[1] "Python wrapper for the thingmagic mercury api." IEEE, October 2019. [Online]. Available: https://github.com/gotthardp/python-mercuryapi

[2] F. Alonso-Martín, J. Gamboa-Montero, J. Castillo, Á. Castro-González, and M. Salichs, "Detecting and classifying human touches in a social robot through acoustic sensing and machine learning," *Sensors*, vol. 17, no. 5, p. 1138, 2017.

[3] A. Ben-Hur and J. Weston, "A user's guide to support vector machines," in *Data mining techniques for the life sciences*. Springer, 2010, pp. 223–239.

[4] Y. Bengio and Y. Grandvalet, "No unbiased estimator of the variance of k-fold cross-validation," *Journal of machine learning research*, vol. 5, no. Sep, pp. 1089–1105, 2004.

[5] W. Chen, L. Chen, Y. Huang, X. Zhang, L. Wang, R. Ruby, and K. Wu, "Taprint: Secure text input for commodity smart wristbands," in *The 25th Annual International Conference on Mobile Computing and Networking*. ACM, 2019.

[6] E. Coyle and E. G. Collins Jr, "A comparison of classifier performance for vibration-based terrain classification," FLORIDA STATE UNIV TALLAHASSEE DEPT OF MECHANICAL ENGINEERING, Tech. Rep., 2008.

[7] K. P. Fishkin, M. Philipose, and A. Rea, "Hands-on rfid: Wireless wearables for detecting use of objects," in *Ninth IEEE International Symposium on Wearable Computers (ISWC'05)*. IEEE, 2005, pp. 38–41.

[8] C. Harrison, J. Schwarz, and S. E. Hudson, "Tapsense: enhancing finger interaction on touch surfaces," in *Proceedings of the 24th annual ACM symposium on User interface software and technology*. ACM, 2011, pp. 627–636.

[9] Z. Ju, J. Wang, and F. Zhu, "Named entity recognition from biomedical text using svm," in *2011 5th international conference on bioinformatics and biomedical engineering*. IEEE, 2011, pp. 1–4.

[10] M. Kaur, M. Sandhu, N. Mohan, and P. S. Sandhu, "Rfid technology principles, advantages, limitations & its applications," *International Journal of Computer and Electrical Engineering*, vol. 3, no. 1, p. 151, 2011.

[11] D. Krevelen and R. Poelman, "A survey of augmented reality technologies, applications and limitations," *The International Journal of Virtual Reality*, vol. 9, no. 2, pp. 1–20, 2010.

[12] G. Laput, C. Yang, R. Xiao, A. Sample, and C. Harrison, "Em-sense: Touch recognition of uninstrumented, electrical and electromechanical objects," in *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM, 2015, pp. 157–166.

[13] C. Li, L. Mo, and D. Zhang, "Review on uhf rfid localization methods," in *IEEE Jorunal of Radio Frequency Identification*. IEEE, 2019.

[14] H. Li, C. Ye, and A. P. Sample, "Idsense: A human object interaction detection system based on passive uhf rfid," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 2015, pp. 2555–2564.

[15] J. Liu, Y. Chen, M. Gruteser, and Y. Wang, "Vibsense: Sensing touches on ubiquitous surfaces through vibration," in *2017 14th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 2017, pp. 1–9.

[16] Y. Liu and Y. Qiu, "An indoor localization of uhf rfid using a hybrid approach," in *2012 2nd International Consumer Electronics, Communications and Networks (CECNet)*. IEEE, 2012.

[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[18] K. V. S. Rao, "An overview of backscattered radio frequency identification system (rfid)," in *1999 Asia Pacific Microwave Conference. APMC'99. Microwaves Enter the 21st Century. Conference Proceedings (Cat. No. 99TH8473)*, vol. 3. IEEE, 1999, pp. 746–749.

[19] K. Rao *et al.*, "Theory and measurement of backscattering from rfid tags," *IEEE Antennas and Propagation Magazine*, vol. 48, no. 6, pp. 212–218, 2006.

[20] A. P. Sample, D. J. Yeager, P. S. Powledge, A. V. Mamishev, and J. R. Smith, "Design of an rfid-based battery-free programmable sensing platform," *IEEE transactions on instrumentation and measurement*, vol. 57, no. 11, pp. 2608–2615, 2008.

[21] A. Schmidt, H.-W. Gellersen, and C. Merz, "Enabling implicit human computer interaction: a wearable rfid-tag reader," in *Digest of Papers. Fourth International Symposium on Wearable Computers*. IEEE, 2000, pp. 193–194.

[22] J. F. Tressler, S. Alkoy, and R. E. Newnham, "Piezoelectric sensors and sensor materials," *Journal of electroceramics*, vol. 2, no. 4, pp. 257–272, 1998.

[23] S.-C. Yu, "Rfid implementation and benefits in libraries," *The Electronic Library*, vol. 25, no. 1, pp. 54–64, 2007.