UIUC - CS 437 - Internet of Things

Capstone Project Report

Rodrigo Mendoza - rm36
Iziren Okhamafe - iokha2

Github link: https://github.com/rm36/IoTCapstoneProject

Video demo links:
https://drive.google.com/file/d/1wHe3e7CmoKpoRzvzIK21arMXvs5dIT5I/view?usp=sharing and
https://drive.google.com/file/d/14FPHZa1Pzr3UHVO6mTJ9mclS0CCxDL8F/view?usp=sharing

Motivation

In our modern day and age, we as human beings can live quite strenuous and busy lives. Often, sometimes coming back from a hard day of work/school could result in some absent mindedness. From time to time, we may forget where we place our car/house keys, wallet, glasses, or other forms of materials that we deem important. The inspiration for this project pertains to how we humans live such complex and complicated lifestyles in this modern world. Due to this, there are bound to be lapses in human self-awareness (for instance, where did I place my car keys? Someone may have poor vision and could have mis-placed their glasses. Thus, this project was to help fulfill the need(s) of those whom may run into conundrums sometimes locating their belongings.

From
https://www.prnewswire.com/news-releases/lost-and-found-the-average-american-spends-25-days-each-year-looking-for-lost-items-collectively-costing-us-households-27-billion-annually-in-replacement-costs-300449305.html , The Average American Spends 2.5 Days Each Year Looking For Lost Items Collectively Costing U.S. Households $2.7 Billion Annually in Replacement Costs. When you look at these expenses, that is quite a significant amount.

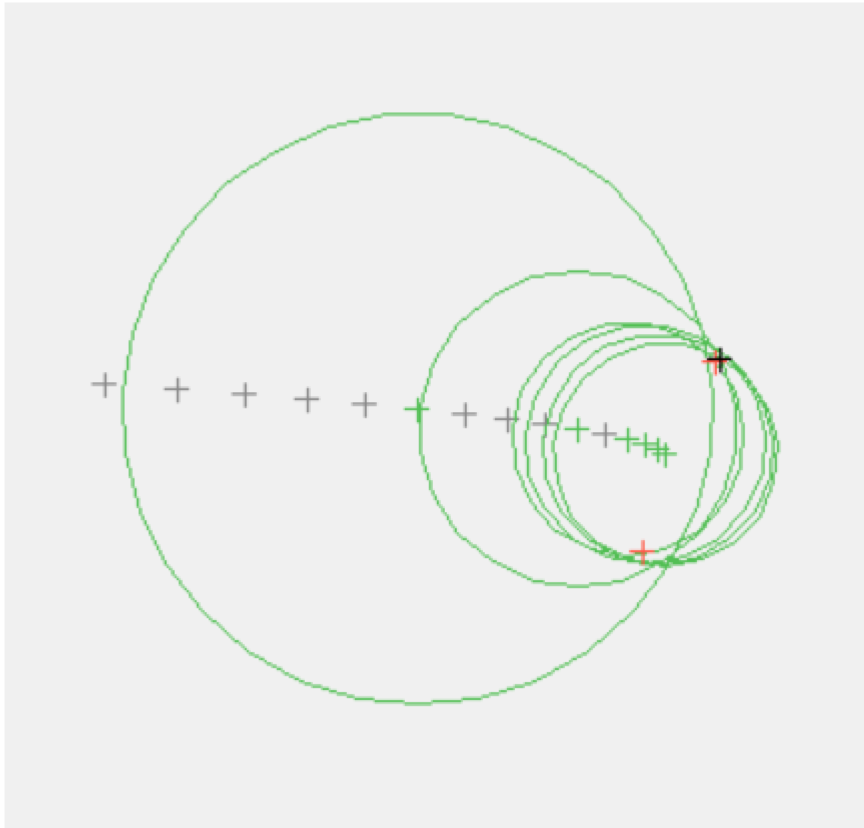
We would like to eliminate the need for people worldwide taking such a significant amount of time searching for their respective items.

Technical approach

The technical approach and implementation details are also explained in conjunction with the code in the explanation video at https://drive.google.com/file/d/1wHe3e7CmoKpoRzvzIK21arMXvs5dIT5I/view?usp=sharing.

We worked in two phases:

- First we validated the trilateration algorithm in order to get a sense of how robust our solution would be in real-world scenarios. The validation was a simulation in python which included simulated RSSI measurements every step, and simulated position of the user moving over time. In the simulation below, the user's position changes over time in the gray crosses, and the green crosses are the data selected for trilateration. Each green circle represents the possible locations of the beacon given a perfect RSSI signal, so the intersection of all the circles should theoretically be at a single point. However in practice, there were two clusters, and only one was close to the beacon.
    - In theory, with 3 circles intersecting at a single point, there should only be one solution, but implementing that algorithm turned out to return inaccurate results in practice, so we instead did clustering of multiple possible points, in the same spirit as the K-means algorithm, and returned the average centroid of the most significant cluster, which represents the beacon's location.
    - We used multiple libraries to prototype, for example Bleak for rapid prototyping of BLE signal measurements.



- In the front-end implementation, the cluster close to the beacon turned out to simply be the first solution to the circle intersection problem, which was much easier to implement than in the simulation.
    - We tried using BLE, but the signal wasn't as accurate compared to Bluetooth, despite the use of lower energy.
    - The server in the Raspberry PI used an advertisement of Bluetooth to get ready to connect, but in our implementation we didn't connect, hence optimizing the power consumption.
    - We chose ARCore as the platform for which to track the position of the user in order to compute the delta and point the arrow towards the estimated solved position of the beacon.

In order to build the device, we decided to get a microcontroller (Raspberry Pi, but others like Arduino and TI Microcontrollers would work as well). Python was the language of choice and an app was built on Android with ARCore to pinpoint the location of the respective device that is lost transmits UHF (ultra high frequency) radio waves to the Android device.

The Android device uses AR (Augmented Reality) to locate the lost device (please see video that illustrates this here: https://drive.google.com/file/d/14FPHZa1Pzr3UHVO6mTJ9mclS0CCxDL8F/view?usp=sharing).

Implementation details

As mentioned above, all of the code can be found in https://github.com/rm36/IoTCapstoneProject.
The detailed explanation of the code can be found in the video as well (https://drive.google.com/file/d/1wHe3e7CmoKpoRzvzIK21arMXvs5dIT5I/view?usp=sharing).

From the literature of signal attenuation it's well known that the signal lowers in a logarithmic scale. The formula we used to obtain the distance, d, from the RSSI was the following[1]:

$$d = 10^{\frac{1mRSSI - RSSI}{10\,N}}$$

Where 1mRSSI is the signal value of the RSSI when the beacon was at 1m of distance from the phone, and N is a constant depending on the attenuation of the signal. In the case of Bluetooth it's around 2, and in our trial and error we found 2.3 to be particularly useful. However, it's a value that needs calibration for different environments. Possibly, this could happen during setup for users of this application.

We used two trilateration algorithms to find the location of the beacon based on the signal:

- The theoretical exact solution for the intersection of 3 circles that should meet at a single point.
- The cluster of points of the intersections between two circles. Each set of two circles (two radii of possible solutions from RSSI signal values) has two intersection points, but in general they cluster in a single point, which is much more resilient than the theoretical exact solution for the intersection of 3 circles. We considered using K-means for clustering but

[1] Formula based on https://blog.estimote.com/post/177348177680/estimote-lte-m-beacon

it turns out that simply choosing the first solution is enough in practice, although not completely reliable for all conditions, for example, a user walking away from the first position estimate of the beacon.

The yaw angle was determined by using the following equation:

$$^{camera}T_{beacon} = {^{World}T_{camera}}^{-1} * {^{World}T_{beacon}}$$

And then, using T as the beacon-to-camera transformation, we compute the yaw angle (in radians):

$$yaw = \frac{\pi}{2} - atan2(-T.z(), -T.y())$$

Completely ignoring the X component of the position component of the transformation, because the height location of the beacon is ignored. The orientation component of the transformation was also ignored because it's not actually estimated from the trilateration.

Results

The project went extremely well! The AR UI works flawlessly and it is able to pinpoint the lost device via bluetooth connectivity. If someone attaches a bluetooth module (like a bluetooth pin to their glasses lens or a pin attached or placed within the wallet) they would be able to locate their device within the confines of their home/apartment.

One must keep in mind that bluetooth operates and omits high frequency waves therefore, its respective wavelength can be quite short. That means over long distances, this technology would not be as effective. However, this device is not meant to work at long distances. It is meant for those who lose their devices at home (e.g. under the couch, on a countertop, etc.). One could easily however implement a version of the device in which belongings over long distances can be tracked when attached to the device. Using GPS modules would be a welcome addition to make this device a device that one could track over long distances granted that it is still attached to the human belonging (wallet, keys, etc.).

Everything went well on the project and nothing went wrong. The reason why the results are important is due to the fact that a working device means that after some more arduous tests are run we can simply release it to the market.

Skill-building

The projects did indeed allow the group members to extend our respective skill sets. New technologies were definitely worked upon. Rodrigo seems to be quite good with Augmented reality. Iziren learned a lot in regards to Augmented reality and Android development in Kotlin. We both experimented more with bluetooth beyond the lecture videos and found how effective it could be when used as a tracking mechanism. Wireless solutions in general can be quite noisy and not 100% accurate. Bluetooth low-energy was found to be better.

Innovation

In a sense, this project is not necessarily “new” and builds up on ideas that were previously implemented. However, the use of Augmented Reality adds an edge to the existing location solutions, as it incorporates an exact estimation of the user’s phone location, and most Android phones include these sets of libraries. As one can see here, there is a bluetooth tracking product for that is made specifically for keychains known as the Tile Mate: https://www.amazon.com/dp/B09998MBFM it transmits bluetooth radio waves up to 250 ft (76 meters).