Positioning using GPS and iBeacons

Henrik Frank, hefra13@student.sdu.dk

Christian Arentsen, chare13@student.sdu.dk

31-03-2017

I. Introduction

Many approaches to in- and outdoor location fingerprinting exist. The aim of this paper is to use two fingerprinting methods, Bluetooth using Bluetooth Low Energy devices and the Global Positioning System to pin point locations on smart devices.

II. Problem statement

Global Navigation Satellite System (GNSS), also known as the Global Positioning System (GPS), is the most well known location system. With a fix of at least 4 satellites a GPS device can pin point the users position within a radius of less than 10 meters. Using additional features of the GPS, the location can be pin pointed within few centimetres. This is true for the outdoors, but since GPS signals are rather weak radio signals on different frequencies, they suffer in cloudy weather and even worse, for indoors location tracking, this is called signal attenuation.

The GPS is great for outdoor positioning, but it is by far the best choice for indoor location fingerprinting. When it comes to indoor fingerprinting, different approaches has been suggested, one of them being Bahl etȧl [1], whom suggest location fingerprinting using Wireless LAN (or WLAN). Their idea is to have enough access points (AP) and knowledge about their location, such that you can calculate signal strength of the AP and from this, derive the location.

Another approach is the use of Bluetooth Low Energy (BLE) devices, such as Apples iBeacon. In the book by LaMarca [2] they discuss the use of Bluetooth, raising the issue of its limited range (approximately 10 meters to a maximum of 25 meters). Regardless of said limits, BLE devices offer a great opportunity to track users within buildings, and University of Southern Denmark has an ongoing project in which they have mounted BLE devices inside a few buildings.

The intent of this short paper is to use BLE devices and GPS, more precisely; to do location fingerprinting using GPS as the main method, and BLE when GPS is not available.

III. Solution proposal

In the problem statement section, we shortly introduced our solution proposal. The goal is to create a prototype Android application, capable of utilizing GPS signals and to search and determine indoor location from the BLE devices placed in building O44 on the University of Southern Denmark.

Our prototype consists of two main components, the Google Maps API and the Kontakt.io API, wrapped in a Drawer menu application, as seen in Figure 1. As this prototype utilizes both GPS and Bluetooth, a bunch of permissions has to be given by the user, thus the user is prompted upon start if these permissions are not given by default, as seen in Figure 2.

After the permissions has been given, also upon start, a background service is tarted, continuously looking for BLE devices, using the Kontakt.io API. This API gives the prototype the

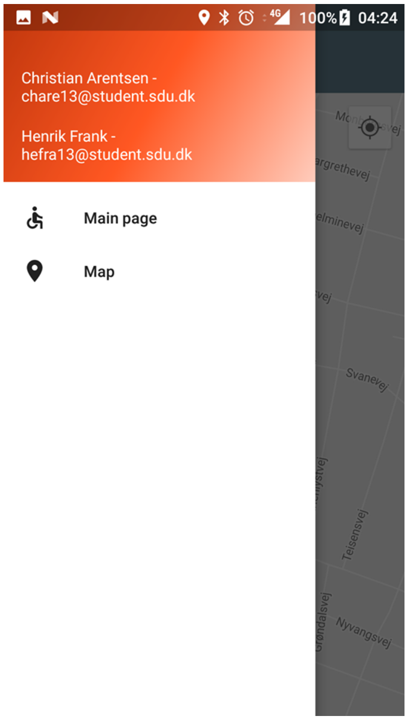
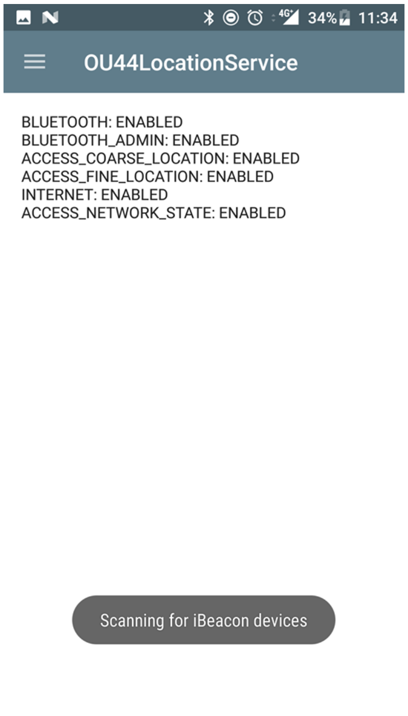


Figure 1: Drawer menu to navigate between functionality



```
BLUETOOTH: ENABLED
BLUETOOTH_ADMIN: ENABLED
ACCESS_COARSE_LOCATION: ENABLED
ACCESS_FINE_LOCATION: ENABLED
INTERNET: ENABLED
ACCESS_NETWORK_STATE: ENABLED
```

Scanning for iBeacon devices

Figure 2: Permissions required to use the prototype, checked upon start

functionality to search for BLE devices (which Android as default has trouble with). If such device is found, the prototype will switch to the Bluetooth navigation screen, from either the main- or the GPS screen.

This means, as the user navigates outside, the user would only benefit from using GPS positioning in Google Maps, as seen in Figure 3, yet, when a BLE device has been discovered, the user has most likely moved indoors due to the low range of Bluetooth as we discussed in the previous sections. Therefore, as soon as the BLE device has been discovered, the prototype will change to the indoor navigation view, displaying the whereabouts of the user, as seen in Figure 4. Currently the BLE positioning is done by measuring all nearby BLE devices and finding the closest, as seen in Listing 1

Listing 1: Code snippet of the BLE positioning

```
public void onIBeaconDiscovered(IBeaconDevice ibeacon,
        IBeaconRegion region) {
    Context context = getApplicationContext();

    int duration = Toast.LENGTH_LONG;

    room_content = (TextView) findViewById(R.id.room_content);

    if(!AppData.getInstance().checkForDevice(ibeacon)){
        AppData.getInstance().addIBeaconDevice(ibeacon);
```

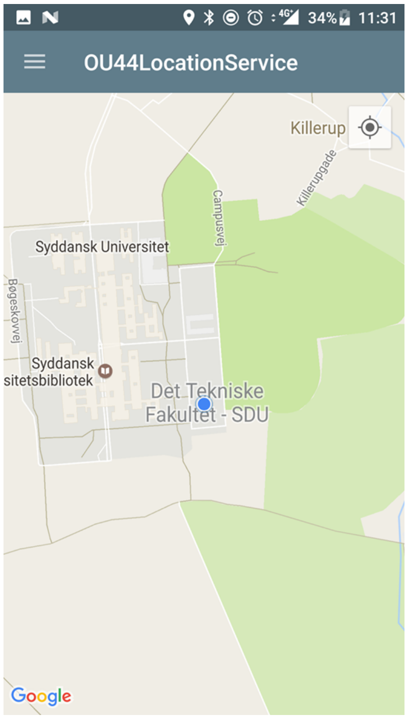
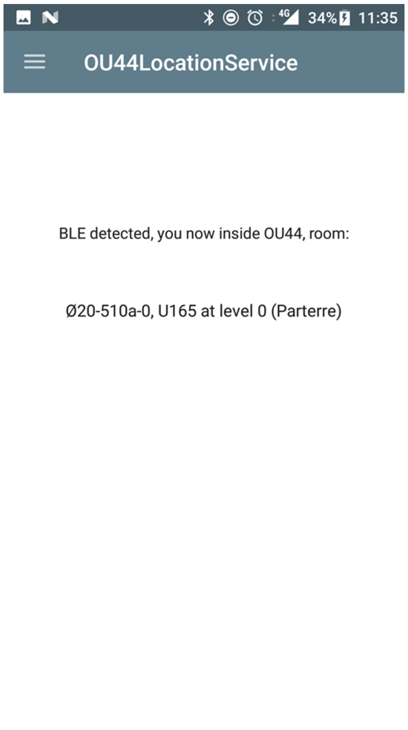


Figure 3: Google Maps navigation.



BLE detected, you now inside OU44, room:

Ø20-510a-0, U165 at level 0 (Parterre)

Figure 4: Bluetooth navigation screen, showing current location

```
          }

          if(!AppData.getInstance().getiBeaconDevices().isEmpty()){
          BeaconItem bi =
          AppData.getInstance().getBeaconData(AppData.
          getInstance().getIBeaconDeviceWithHighestRSSI());
          if(bi != null){
                    IndoorFragment indoorFragment = new IndoorFragment();
               if(room_content != null){
               room_content.setText(bi.getRoom() + ",␣" + bi.getRoomName()
               + "␣at␣level␣" + bi.getLevel());
          }
          getSupportFragmentManager().beginTransaction().replace(
          R.id.content_fragment_container,indoorFragment).commit();
          }
}
}
};
```

IV. Conclusion

This paper introduced an android prototype used to position users by switching between GPS for outdoor positioning and Bluetooth Low Energy devices for indoor positioning. By doing so, we overcome the issues of indoor tracking, as this is not possible with GPS. The indoor tracking is done by measuring the shortest distance to all nearby BLE devices, and selecting this as the current location, giving the user an idea about his or hers location in a given building.

When using BLE in order to positioning, more BLE devices should be provided, but also more data associated with the device is neccesary. The precision is limited to only letting the user know which room it is located in, but not the excact coordinate. In order to provide this information, more than 4 devices must be provided, which each contains information about its own location, as these can be used to triangulate the user. Even then the precision is low, as the triangulation is based on the RSSI power given by the BLE device. The ammount of power reached to the user can vary very much from time to time. Power consumption and remaining battry life of the BLE device can also affect the performance of RSSI output.

As a final remark it must be noted that this prototype does not deal with energy consumption, though the Kontakt.io API is rather energy efficient already.

References

[1] Paramvir Bahl and Venkata N. Padmanabhan. Radar: An in-building rf-based user location and tracking system. *IEEE INFOCOM*, 0-7803-5880-5/00, 2000.

[2] Anthony LaMarca and Eyal de Lara. Location systems: An introduction to the technology behind location awareness. *Synthesis Lectures on Mobile and Pervasive Computing*, 3(1):1–122, jan 2008.