BLuEye

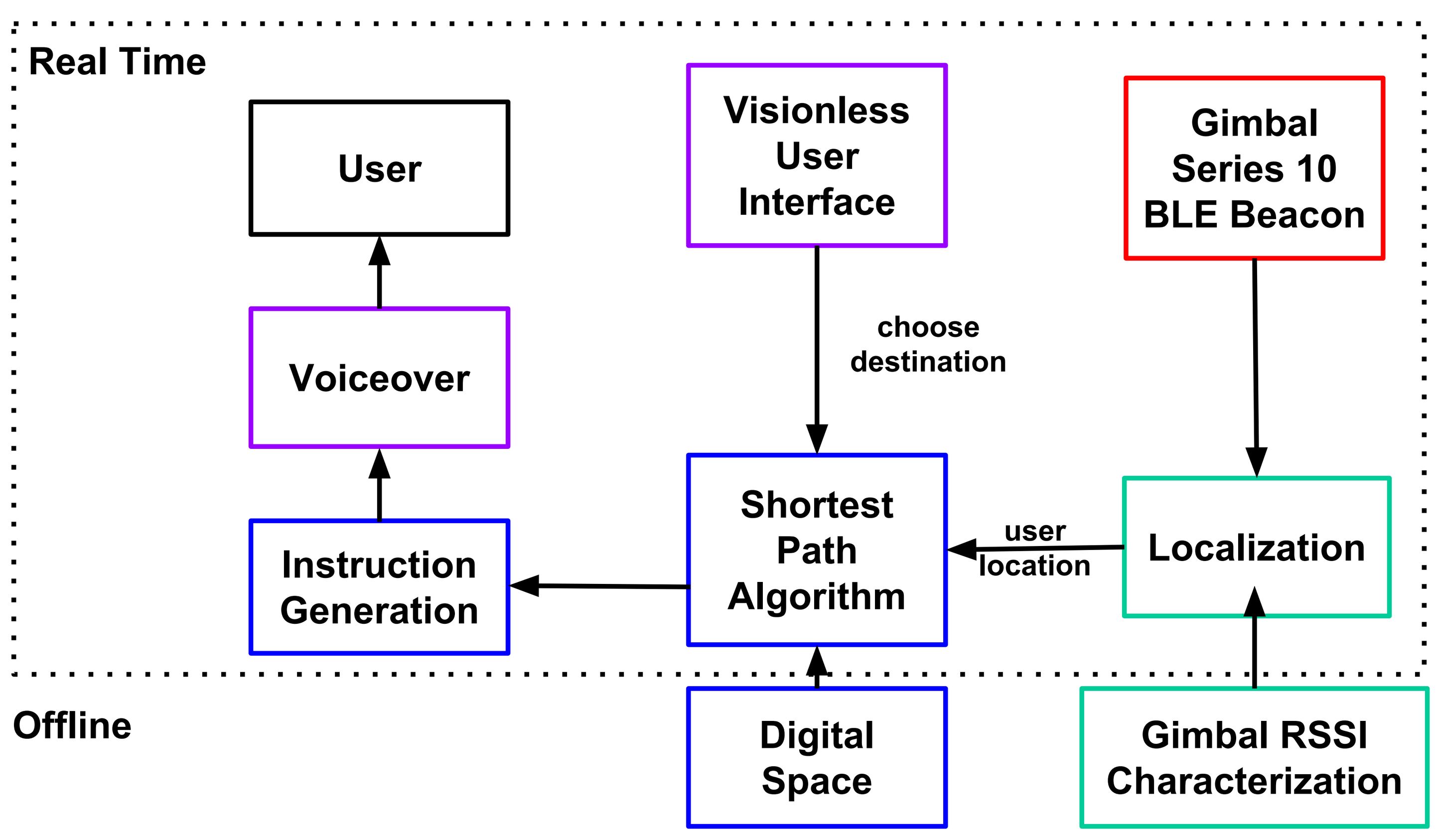
Krista Lohr, Steve Fialli, Divya Reddy, Thomas Kelly
Faculty Advisor: Prof. Aura Ganz

Abstract

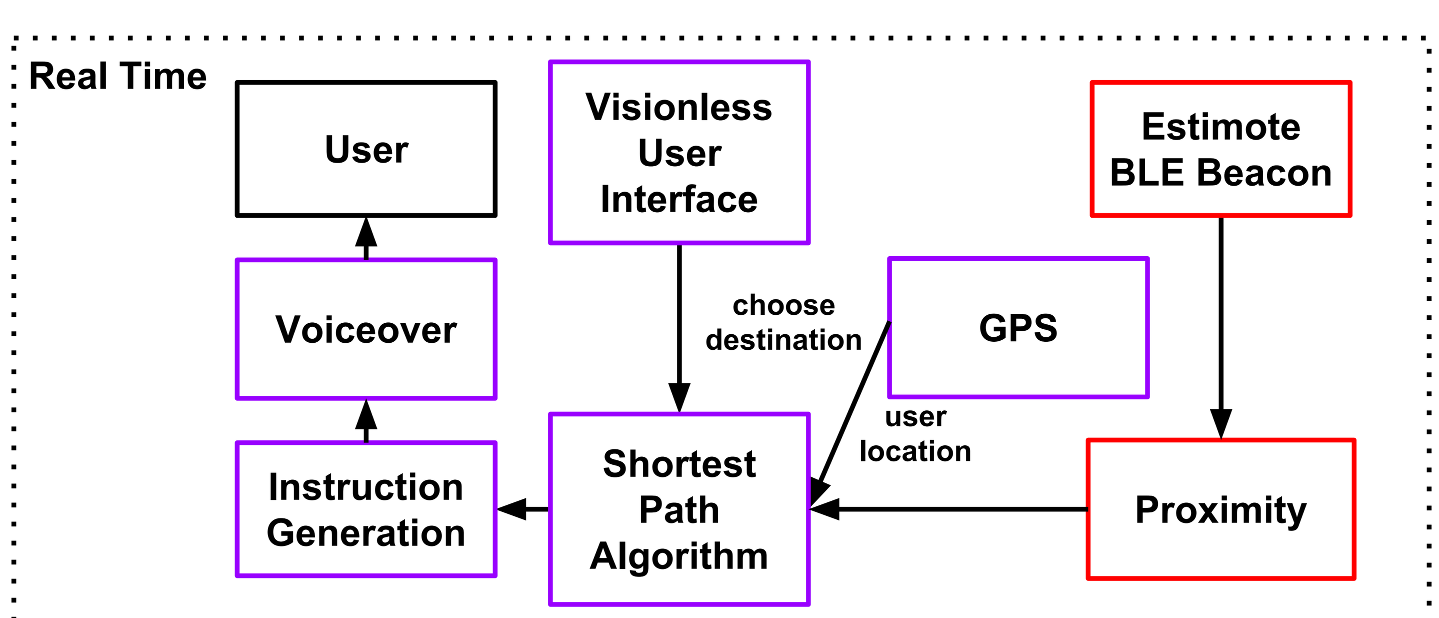
BLuEye is a navigation system that will guide the blind and visually impaired in unfamiliar indoor and outdoor environments. The system utilizes a mobile application that communicates with Bluetooth Low Energy beacons. The system can establish user location and provide voice instructions for guidance to a specified destination. The system will notify the user when they have reached their destination or several other predefined landmarks. The system will also notify the user when they are moving away from their outdoor destination.

Block Diagram

Indoor Block Diagram



Outdoor Block Diagram

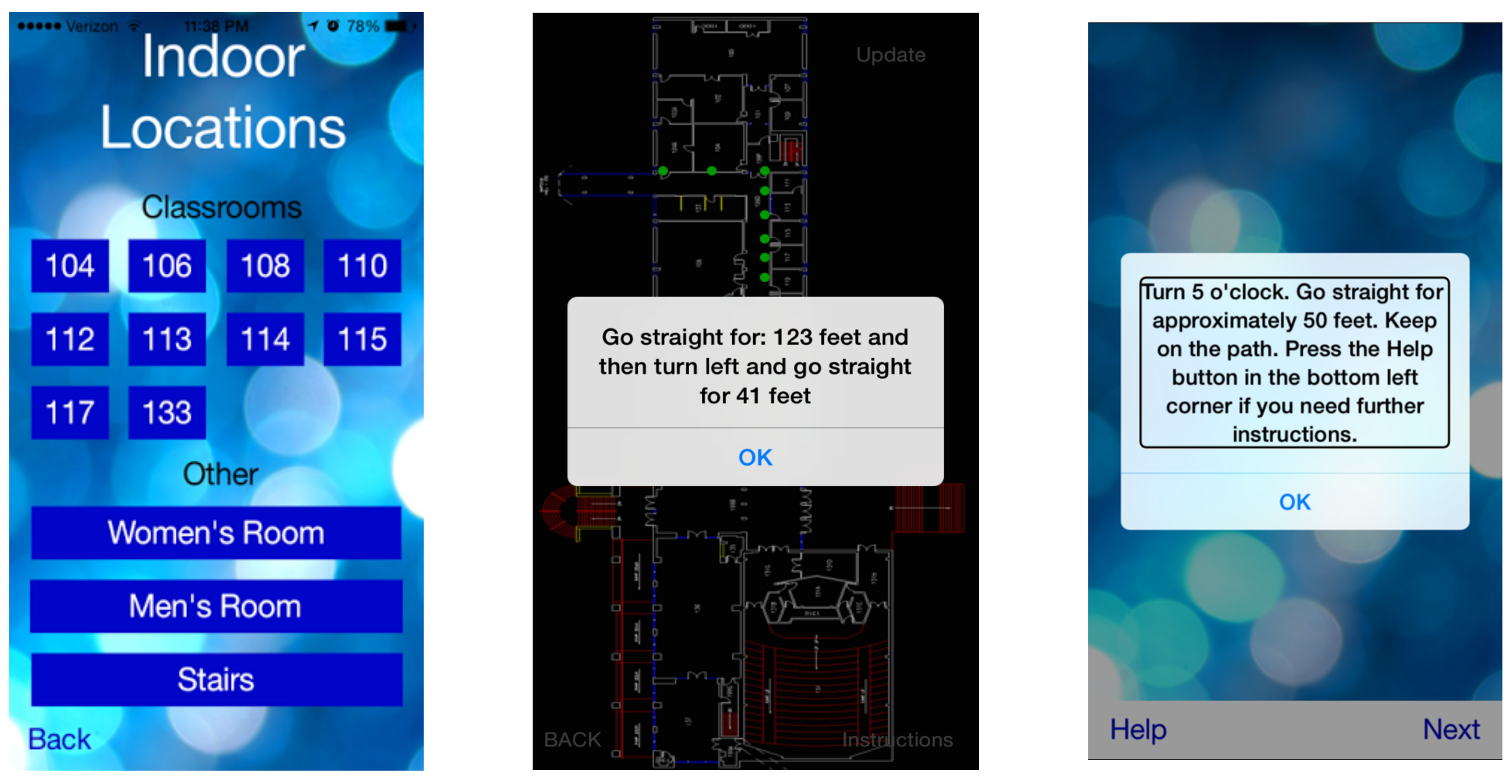


Specifications

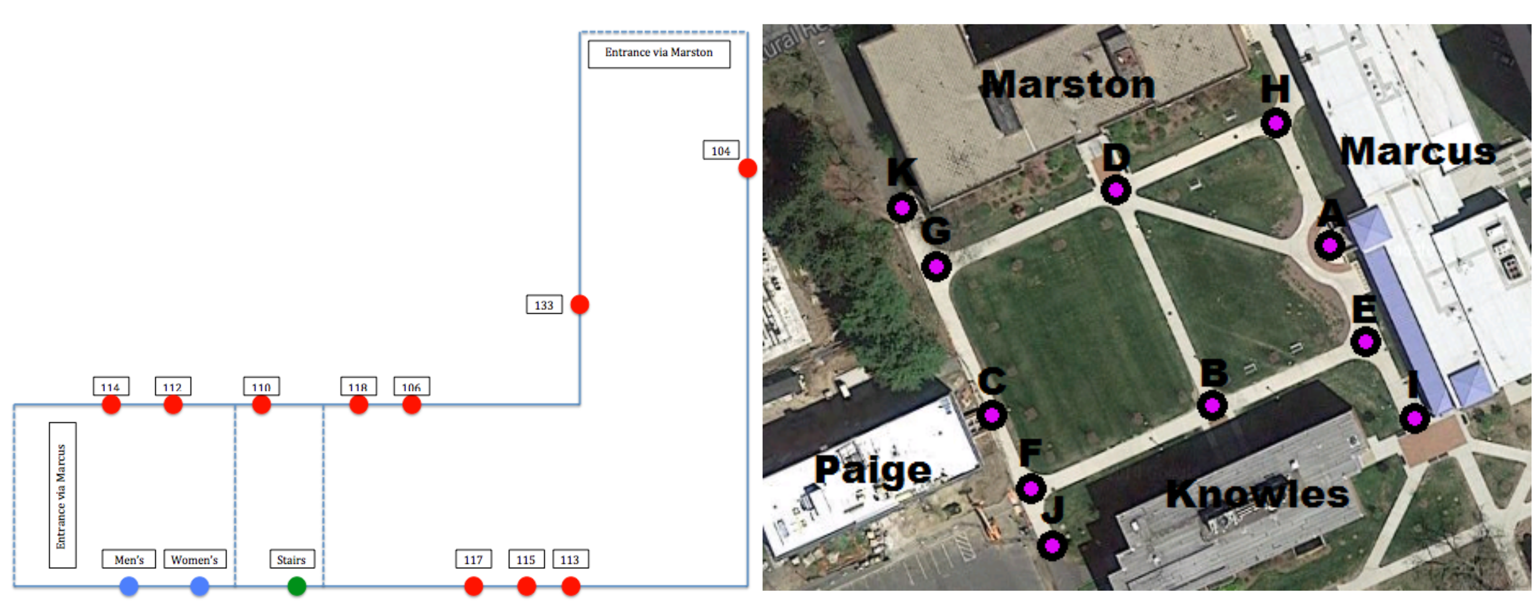
- Successful navigation without visual information
- Localization estimation accurate to 5 feet
- Continuous beacon coverage of entire Engineering Quad (no dead zones)
- Easy to understand and sufficiently frequent voice instructions
- Compatible with Apple devices
- Low maintenance (long lasting power source, weather resistance)

System Overview

BLuEye uses a mobile device application running on the Apple iOS platform. The Graphical User Interface (GUI) was designed to be compatible with the accessibility program, Voice Over.



The indoor system determines the user's location and shortest path to their destination using BLE beacons. The outdoor system uses GPS to do the same. Instructions are generated to help the user navigate. BLE beacons are used to transition between the outdoor and indoor environments. BLuEye will notify the user when they have reached their destination.



Results

- System provides instructions with sufficient accuracy to guide user to destination
- System notifies user when they are approaching destination
- System notifies user when they are walking away from outdoor destination
- Tests conclude that the system can be used to get to destination without any visual information

Acknowledgement

Special Thanks to Professor Aura Ganz for her excellent advising. Thank you to Professors William Leonard and Ramakrishna Janaswamy for helpful evaluation and feedback. Thank you to COO Professor Christopher Hollot and CTO Professor Christopher Salthouse. Also Thanks to Fran Caron for ordering parts.

RSSI Characterization

The BLE beacons transmit signals that are received by the mobile device. The power in these signals decrease as they they get farther from the beacon. The approximate distance from the user's device to a beacon can be determined by analyzing the signal strength at the receiver, known as RSSI. The relationship between RSSI and distance is shown in the path loss model below.

$$RSSI[dBm] = -n\log_{10}(d)+A[dBm]$$

RSSI[dBm] = Received Signal Strength Indicator
n= path loss exponent
d = estimated distance (meters)
A[dBm] = RSSI value obtained when tag is 1 meter from device

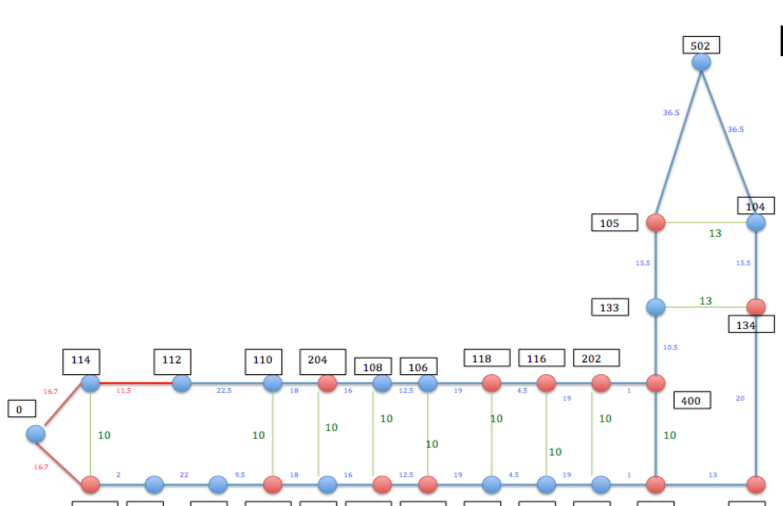
Shortest Path Algorithm

Outdoor

The outdoor system uses two inputs to determine a path to the user's choice destination: coordinates taken from the GPS and the user's choice destination. The shortest path to the destination is determined and the first voice instruction is given to the user. Each voice instruction will guide the user from one node to the next until their destination is reached.

Indoor

The Dijkstra's Algorithm was implemented in the Indoor portion of the algorithm. The inputs will be the user's current location and destination. The user's location was determined via the localization algorithm. The output of the algorithm will be the distance to destination and the path. To construct the navigation algorithm, several nodes had to be predetermined. The nodes were placed at the entrances of classrooms and facilities. Upon running the algorithm ned.



Cost

Development

Part	Price
Gimbal Beacons (x23)	$100
Estimote Beacons (x6)	$200
Developer Account	$100
Total	$400

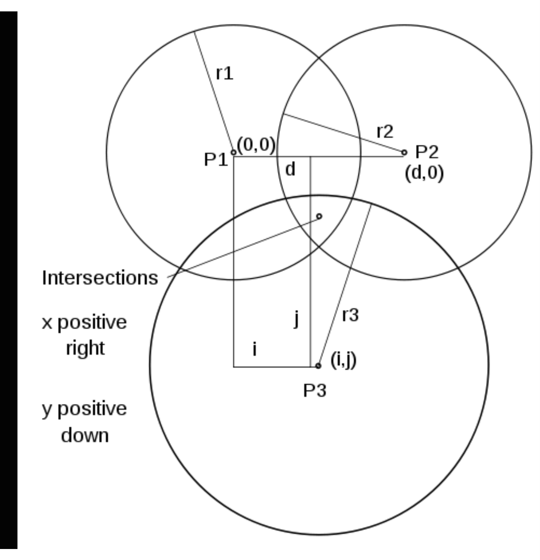
Production

Part	Price
Gimbal Beacons (~20)	$100
Estimote Beacons (~6)	$200
Total	$300

Indoor Localization

To find the users location, we used a trilateration equation based on the distance calculated to the three closest Gimbal beacons in the users range. Our program takes each beacon it sees, converts their signal strength to a distance, sorts all of the beacons by distance, and uses the closest three beacons to find their location.

```
- (CurrentLocation*) positionFinder:(Beacon*) beacona :(Beacon*) beaconb :(Beacon*) beaconc :(double)
   distancea :(double) distanceb :(double) distancec
{
   CurrentLocation *location = [[CurrentLocation alloc]init];
   double va = ((pow(distanceb, 2) - pow(distancec, 2)) - (pow(beaconb.XCoordinate, 2) - pow(beaconc.
      XCoordinate, 2)) - (pow(beaconb.YCoordinate, 2) - pow(beaconc.YCoordinate, 2)))/2;

   double vb = ((pow(distanceb, 2) - pow(distancea, 2)) - (pow(beaconb.XCoordinate, 2) - pow(beacona.
      XCoordinate, 2)) - (pow(beaconb.YCoordinate, 2) - pow(beacona.YCoordinate, 2)))/2;

   double y = (vb*(beaconc.XCoordinate - beaconb.XCoordinate) - va*(beacona.XCoordinate - beaconb.
      XCoordinate))/((beacona.YCoordinate - beaconb.YCoordinate)*(beaconc.XCoordinate - beaconb.
      XCoordinate) - (beaconc.YCoordinate - beaconb.YCoordinate)*(beacona.XCoordinate - beaconb.
      XCoordinate));

   double x = (va - y*(beaconc.YCoordinate - beaconb.YCoordinate))/(beaconc.XCoordinate - beaconb.
      XCoordinate);

   location.XCoordinate = x;
   location.YCoordinate = y;

   return location;


}
```



Once the coordinates of the user are found by this algorithm, they are used to determine the closest node on the path. We can determine direction of the user by comparing this node to their destination node. To determine when the user has reached their destination, we use a combination of localization and proximity.

Instruction Generation

Outdoor

The outdoor system determines instructions using the shortest path algorithm and the heading value taken from the phone. The first instruction will guide the user to the first node, providing directional information and a distance estimated to 5 feet. At each node of the path, a new instruction will be given. If the user walks in the wrong direction, an alert gives them an updated instruction. Estimote proximity is used to give the final instructions when the user is close to their destination building to guide the user to the entrance and into the building.

Indoor

The indoor system calculates the instruction using the Dijkstra's Algorithm. The instructions are divided into sections of the hallway. Each set of instructions will guide the user between sections with a distance estimation. The final instruction will also tell the user what side of the hallway the destination is on. Once the user has reached the proximity of their destination, an alert will tell them that they have reached their destination.

RSSI to Distance