M202A Team Project

Ruoye Wang & Jinchen Wu

Palantir2021

Obstacle avoidance system for the visually impaired

Background

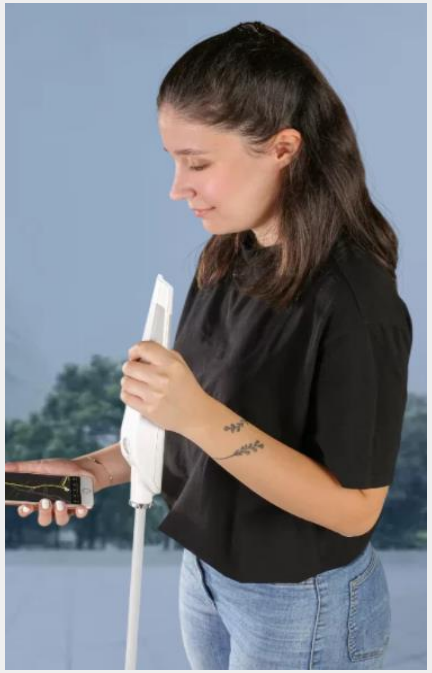
What motivated us?

Guide dog & white cane

- Expensive
- Physical needs
- Limited detection

Wearable devices

- High cost



Smart cane

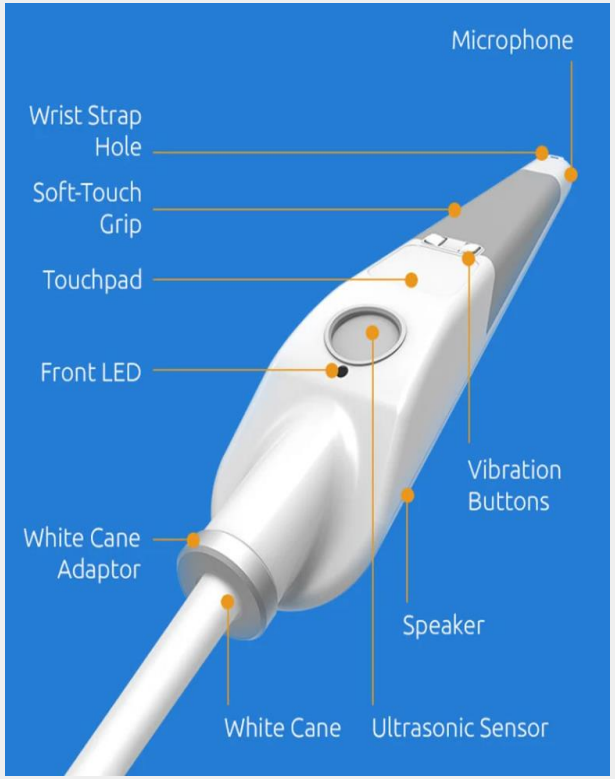
- High cost
- Limited hand mobility

Prior works

What inspired us?

Augmented Cane

- Comprehensive
- Feedback
  - Kinesthetic omni-wheel
  - Audio
  - Push button



WeWalk

- Ultrasonic detection
- Vocal &tactile feedback
- Navigation
- Transportation assistant

Navigation system for blind

- Wearable
- Multiple ultrasonic modules
- GPS navigation
- Vocal and vibrating feedback

Goals

What do we want to achieve?

Robust
- Able to tolerate perturbations and interruptions

Accurate
- Successful detection

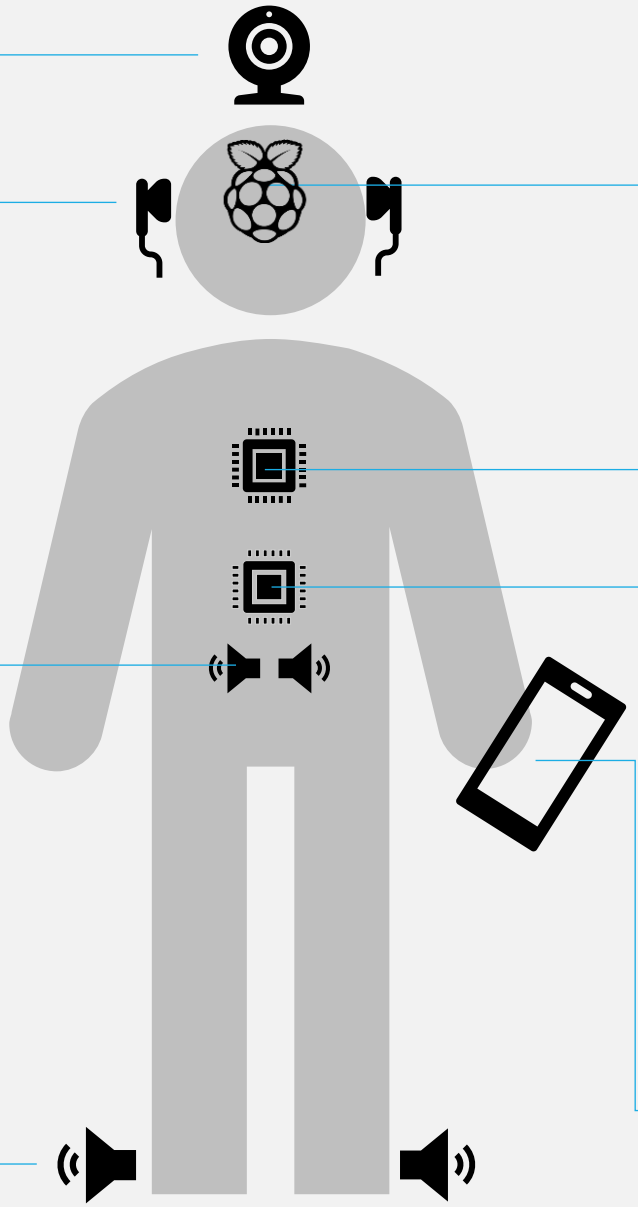
Palantir2021 Obstacle Avoidance System

User-friendly
- Easy to use

Low-cost
- Affordable

Goal

How do we achieve it?



IR Camera

- Preliminary detection of environment

Raspberry Pi

- Process data from IR camera

Earable earphones

- Provide vocal feedback to user

Arduino (Control)

- Overall control of the system
- Send message to smartphone app

Arduino (Sensors)

- Process data from ultrasonic sensors

Ultrasonic Sensors

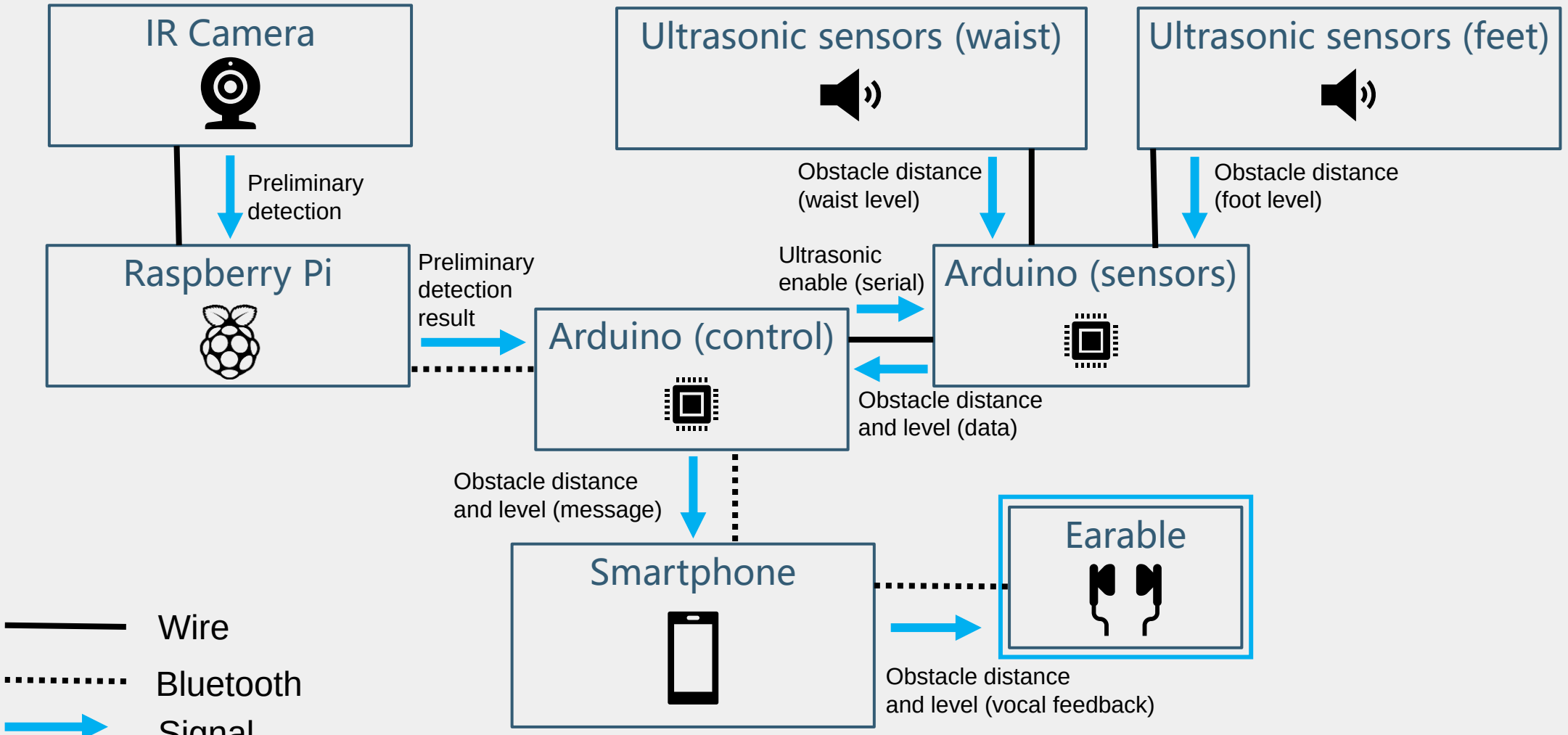
- Detailed detection of obstacles

Smartphone app

- Receive message from Arduino (control)
- Control Earable earphones

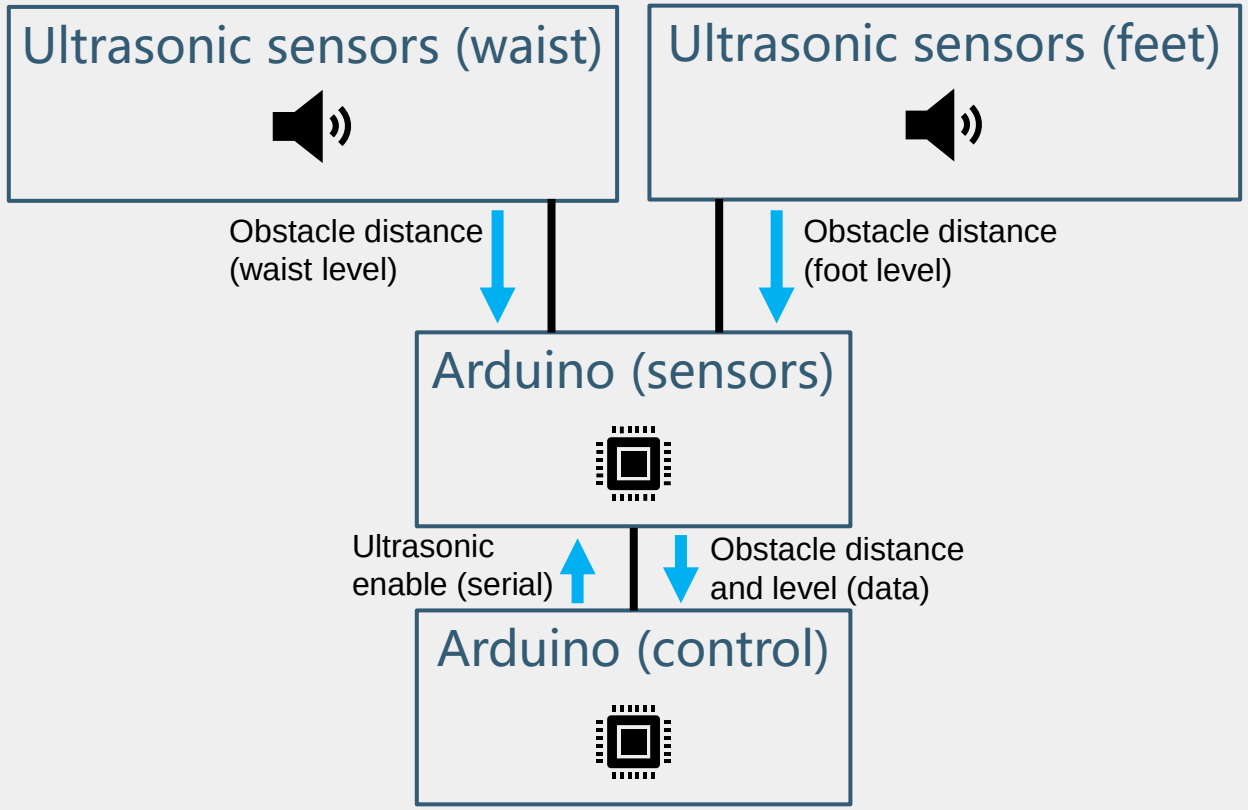
System Overview

Technical approach



Ultrasonic Module

Technical approach



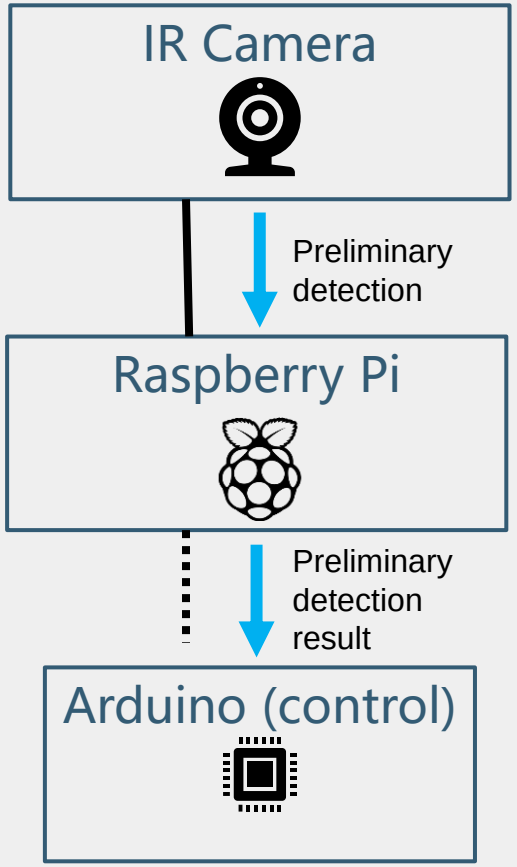
- HC-SR04
- Waist- and foot-level
- Adjustable threshold range
- Serial communication

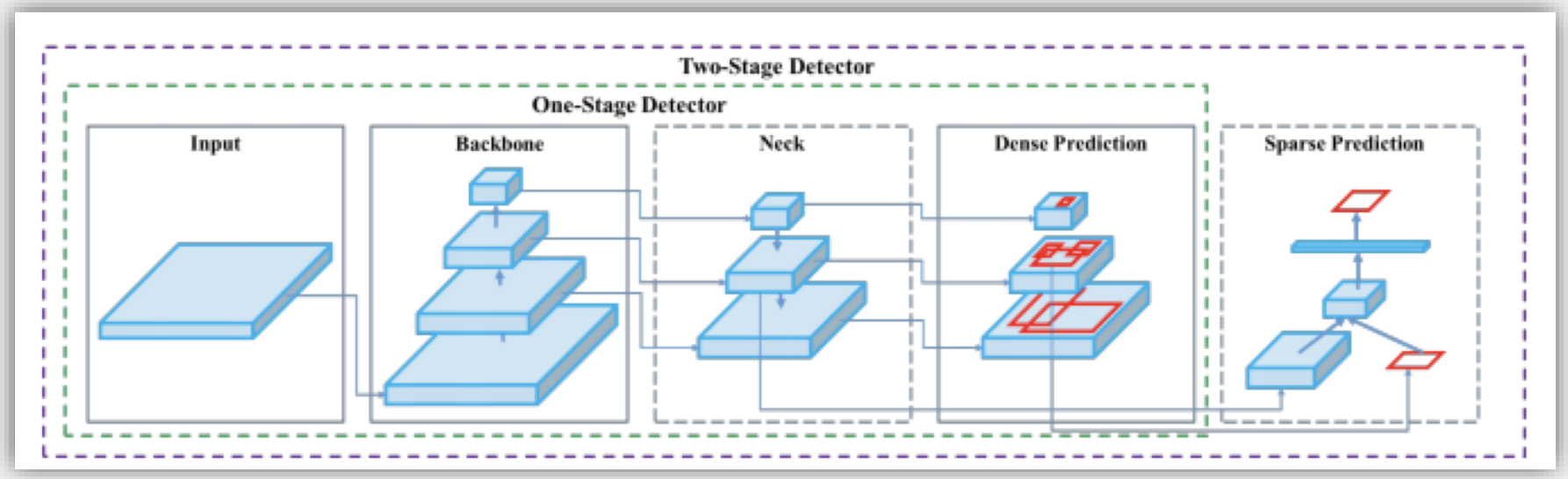
```
void serialEvent() {

  while(Serial1.available() > 0)
 {
   readdata  =  readdata  +char(Serial1.read());
   //delay(2);
 }
   //Everytime sensor Arduino gets something from controller Arduino
   //It conducts a group of detection for left & right waist and left & right foot
   //So it needs controller Arduino to send something continuously
   //Nothing comes from controller, which means there's no need to use ultrasonic modules; sensor Arduino does nothing
   // get enable_detect:
   //Serial.print(readdata);
   if (readdata == "o") {
   /******START DETECTION******/
   /***Left waist***/
   digitalWrite(trig_wl,LOW);
   delayMicroseconds(2);
   digitalWrite(trig_wl,HIGH);
   delayMicroseconds(10);
   digitalWrite(trig_wl,LOW); //Send a 10ms high pulse to trig trig_wr
   left_waist = (pulseIn(echo_wl,HIGH))/58.0;//Receive echoed signal
   /***Right waist***/
   digitalWrite(trig_wr,LOW);
   delayMicroseconds(2);
   digitalWrite(trig_wr,HIGH);
   delayMicroseconds(10);
   digitalWrite(trig_wr,LOW);
   right_waist = (pulseIn(echo_wr,HIGH))/58.0;
   /***Left foot***/
   digitalWrite(trig_fl,LOW);
   delayMicroseconds(2);
   digitalWrite(trig_fl,HIGH);
   delayMicroseconds(10);
   digitalWrite(trig_fl,LOW);
   left_foot = (pulseIn(echo_fl,HIGH))/58.0;
   /***Right foot***/
   digitalWrite(trig_fr,LOW);
   delayMicroseconds(2);
   digitalWrite(trig_fr,HIGH);
   delayMicroseconds(10);
   digitalWrite(trig_fr,LOW);
   right_foot = (pulseIn(echo_fr,HIGH))/58.0;
   delay(100);
   /***SEND DISTANCES***/
   serialWriteDistances();

 }
 readdata = "";
}
```

```
void basicTask(){
   /***Left foot***/
    digitalWrite(trig_fl,LOW);
    delayMicroseconds(2);
    digitalWrite(trig_fl,HIGH);
    delayMicroseconds(10);
    digitalWrite(trig_fl,LOW);
    left_foot = (pulseIn(echo_fl,HIGH))/58.0;
    /***Right foot***/
    digitalWrite(trig_fr,LOW);
    delayMicroseconds(2);
    digitalWrite(trig_fr,HIGH);
    delayMicroseconds(10);
    digitalWrite(trig_fr,LOW);
    right_foot = (pulseIn(echo_fr,HIGH))/58.0;
    delay(50);
    if(left_foot<30||right_foot<30){
      Serial1.print("D");
    }
}
```

Raspberry Pi & IR Camera

Technical approach

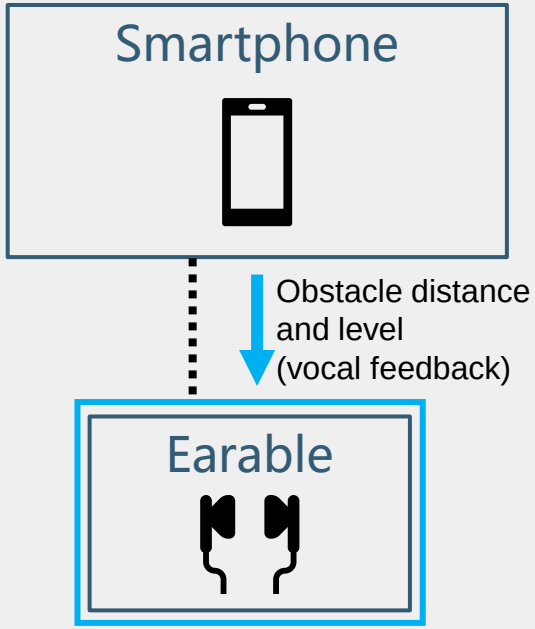




- YOLO algorithm
- Observe in advance
- Peripheral device – Arduino
- Central device - RBPi

```
def SendMsg():
    try:
        p = btle.Peripheral("6d:af:ec:47:07:b8")  #connect to Arduino
        service_uuid = btle.UUID("180C")          #controller Arduino's service uuid
        s = p.getServiceByUUID(service_uuid)
        uuidConfig = btle.UUID("2A56")
        c = s.getCharacteristics(uuidConfig)[0]
        data = c.read()
        if (data != b'\x00'):                     #write signal
            c.write(b'\x00',withResponse=True)
        p.disconnect()
    except BTLEException as exception:
        print("failed")
```

Smartphone Application & Earable Earphones



- Android app as BLE radio
- Phone as BLE peripheral
- Service & characteristic
- 3 priorities of alerts

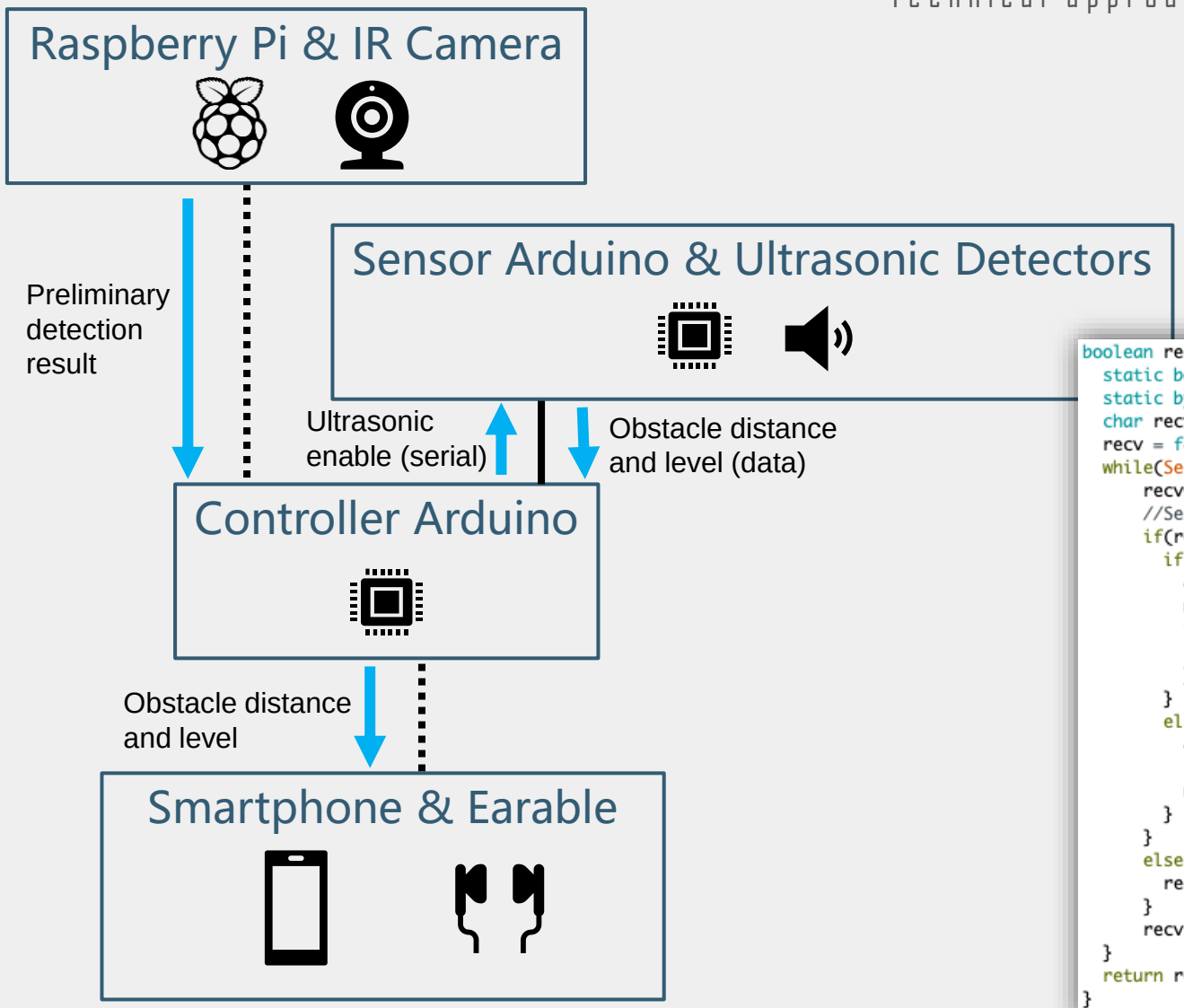
Technical approach

```
private void onResponseToClient(byte[] reqeustBytes, BluetoothDevice device, int requestId, BluetoothGattCharacteristic characteristic) {
    Log.e(TAG, String.format("onResponseToClient: device name = %s, address = %s", device.getName(), device.getAddress()));
    Log.e(TAG, String.format("onResponseToClient: requestId = %s", requestId));
    String msg = OutputStringUtil.transferForPrint(reqeustBytes);
    println("receive:" + msg);
    showText( msg: "receive:" + msg);
    switch (msg){
        case "1": showSound(R.raw.waist);
            showText( msg: "Obstacle at waist detected");
            break;
        case "2": showSound(R.raw.feet);
            showText( msg: "Obstacle at feet detected");
            break;
        case "3": showSound(R.raw.waist);//showSound(R.raw.feet);
            showText( msg: "Obstacle at waist and feet detected");
            break;
        case "4": showSound(R.raw.watchyourstep);
            showText( msg: "Obstacle at feet detected");
            break;
    }
}
```

```
private void initServices(Context context) {
    bluetoothGattServer = mBluetoothManager.openGattServer(context, bluetoothGattServerCallback);
    BluetoothGattService service = new BluetoothGattService(UUID_SERVER, BluetoothGattService.SERVICE_TYPE_PRIMARY);

    //add a read characteristic.
    characteristicRead = new BluetoothGattCharacteristic(UUID_CHARREAD, BluetoothGattCharacteristic.PROPERTY_READ, BluetoothGattCharacteristic.PERMISSION_READ);
    //add a descriptor
    BluetoothGattDescriptor descriptor = new BluetoothGattDescriptor(UUID_DESCRIPTOR, BluetoothGattCharacteristic.PERMISSION_WRITE);
    characteristicRead.addDescriptor(descriptor);
    service.addCharacteristic(characteristicRead);

    //add a write characteristic.
    BluetoothGattCharacteristic characteristicWrite = new BluetoothGattCharacteristic(UUID_CHARWRITE,
            properties: BluetoothGattCharacteristic.PROPERTY_WRITE |
                    BluetoothGattCharacteristic.PROPERTY_READ |
                    BluetoothGattCharacteristic.PROPERTY_NOTIFY,
            BluetoothGattCharacteristic.PERMISSION_WRITE);
    service.addCharacteristic(characteristicWrite);

    bluetoothGattServer.addService(service);
    Log.e(TAG, msg: "initServices ok");
    showText( msg: "initServices ok");
}
```

Overall Control Logic on Controller Arduino

Technical approach

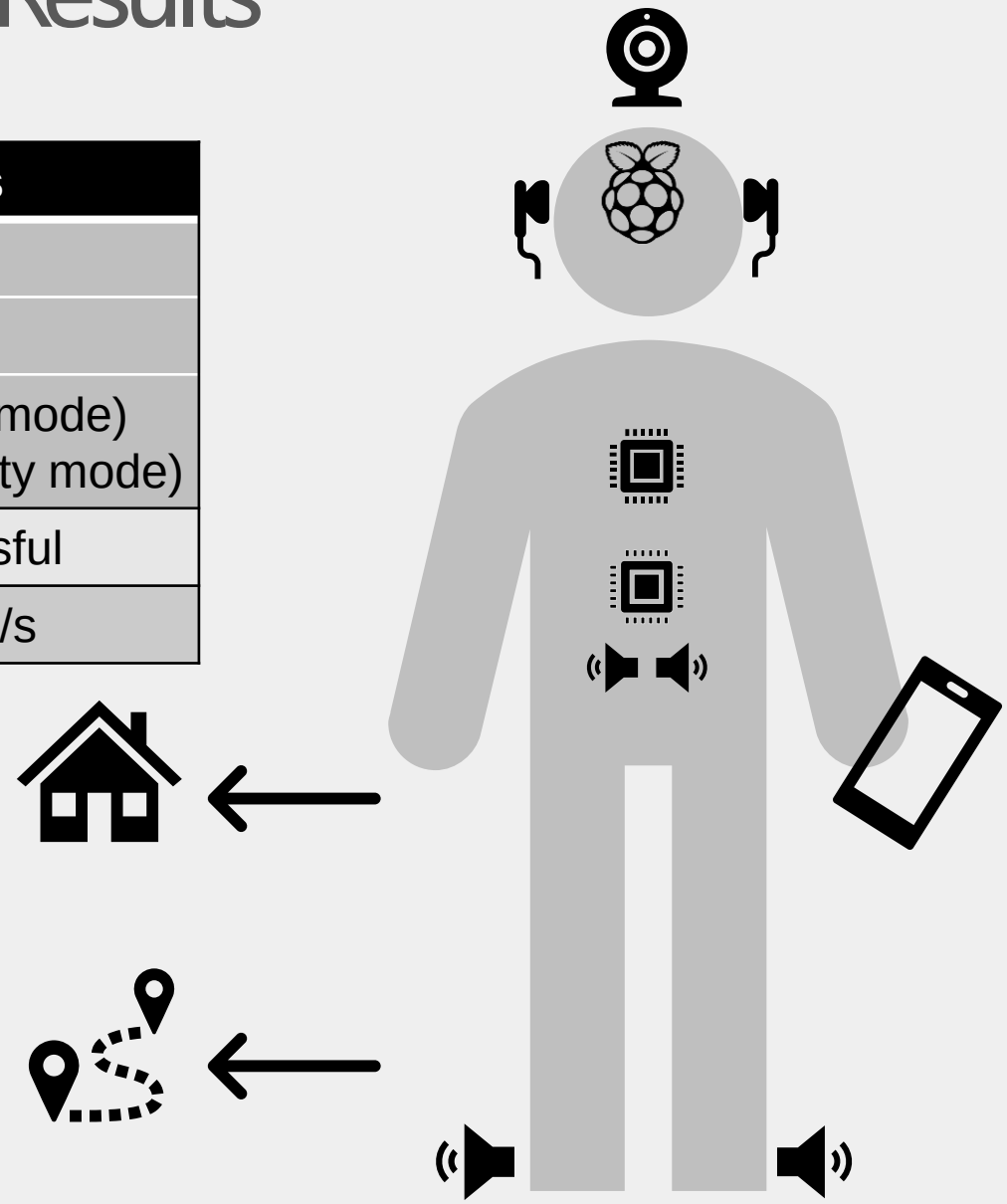



```
void serialEvent() {
  while(Serial1.available()){
    char inChar = (char)Serial1.read();
    if(inChar=='D'){                    //emergency
      //BLE.disconnect();
      notify = 4;
      sendtoPhone();
      notify = -1;
      while(Serial1.available()){
        char tmp = (char)Serial1.read();
      }
    }
    else if( inChar == '<'){ .      //receive distance
      recvUltrasonic();
    }
  }
}
```

```
boolean recvUltrasonic(){
  static boolean recvInProgress=false;
  static byte ndx=0;
  char recvChara;
  recv = false;
  while(Serial1.available()>0){  //When serial is open for communication between Arduino boards
      recvChara=Serial1.read();
      //Serial.println(recvChara);
      if(recvInProgress==true){ //Receive in progress, serial reading not finished
        if(recvChara!=ultra_end){ //If received char is not end marker
          distanceUltrasonic[ndx]=recvChara;  //Read this char into current location of ultrasonic distance
          ndx++;  //Index move to the next location of ultrasonic distance string
          if(ndx>=numUltrasonic){
            ndx=numUltrasonic-1;
          }
        }
        else {  //received char is end marker
          distanceUltrasonic[ndx]='\0'; //Indicating the end of string
          recvInProgress=false;  //receive not in progress, serial reading finished
          ndx=0;  //reset index to 0
        }
      }
      else if(recvChara==ultra_start){  //receive not in progress, but serial reads start marker
        recvInProgress=true;  //so start a receiving progress
      }
      recv = true;
  }
  return recv;
}
```

Evaluation and Results

Metrics	Results
Obstacle detection time (camera)	4s
Warning reception time (app)	< 4s
Total time (obstacle enters range → user receives a warning)	6~9s (normal mode) 3~4s (high-priority mode)
Obstacle Detection Success Rate	All successful
User Movement Speed	0.2~0.3m/s



- Different obstacles in test groups
- Manual calculations

Strengths & Weaknesses

Discussion

S1

Trade-off point

- Relatively low cost
- Satisfying accuracy

S2

Convenience

- Not occupying hand
- Wearable

S3

User-friendly

- Smartphone application
- Adjustable parameters

W1

Performance

- Image processing speed

W2

Hardware connection

- Jumpers and breadboards

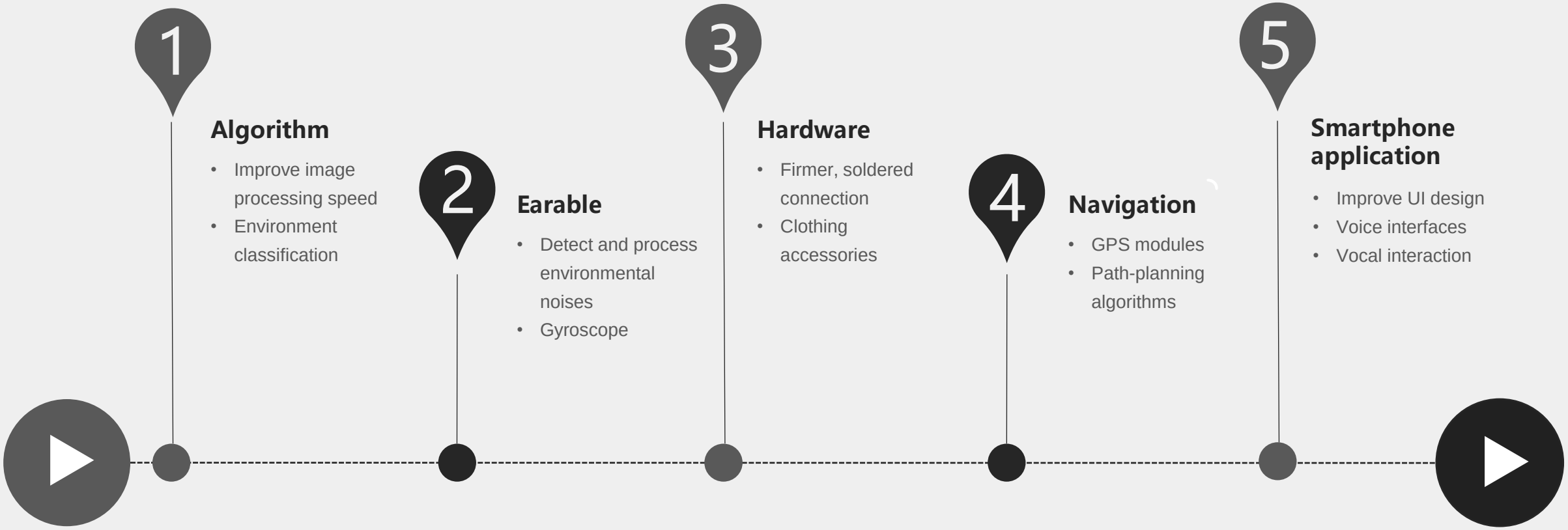
W3

Smartphone application

- Small button
- Without vocal interface

Future Directions

Discussion



Conclusion

Palantir2021

- Obstacle avoidance using IR camera, Raspberry Pi, YOLO, ultrasonic sensors and Arduino

Performance

- High accuracy, satisfactory speed

Future works

- Algorithmic optimization, Earable, hardware connections, navigation, UI re-design

References & GitHub Address

GitHub repository

https://github.com/Palantir2021/ecem202a_project

GitHub pages

https://palantir2021.github.io/ecem202a_project/

Technical references

Arduino https://forum.arduino.cc/t/serial-input-basics-updated/382007

YOLO https://github.com/ultralytics/yolov5/tree/8f354362cd94c70908bf6168951b07bd32715ebe;

https://blog.roboflow.com/yolov5-improvements-and-evaluation/

Prior work inspirations

Augmented Cane P. Slade, A. Tambe, M. J. Kochenderfer, Multimodal sensing and intuitive steering assistance improve navigation and mobility for people with impaired vision. Sci. Robot. 6, eabg6594 (2021)

WeWalk https://wewalk.io/en/

Navigation System for Blind https://www.youtube.com/watch?v=8tbeELHKx9Y; https://www.youtube.com/watch?v=JMl07-yIUo8&t=149s

Thank you

Jinchen Wu | 305627306
Ruoye Wang | 605625594
Dec. 12, 2021