

ME GY-6933 Advanced Mechatronics: Raspberry Pi Project

Cameye

Kishan Gondalia, Harish Kuppam, Krishnesh Meratia

Abstract

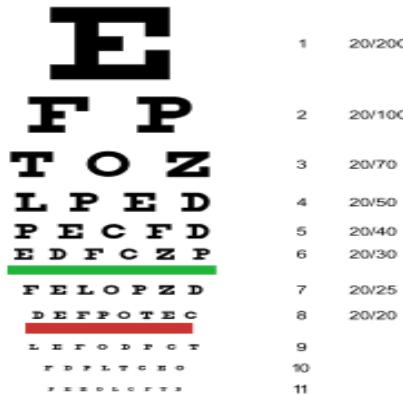
Visual impairment is a compelling issue for our society. As people age they are affected by this debilitating condition. Technological progress offers unique means to improve life conditions for the visually impaired. Here we present an approach to help them by exploiting the computational power of raspberry pi and the efficient packages available in it for computer vision. Cameye is a device prototype for helping the impaired to help in mobility by identifying objects and helping them localize. Actuators are used to alert the person of nearby objects helping from collision controlled by an Arduino.

Background

Visual impairment is a decreased ability to see to a degree that causes problems not fixable not usual means such as glasses. The term blindness is used for complete or nearly complete vision loss. Visual impairment may cause difficulties with normal daily activities such as driving, reading, socializing, and walking. The most common causes of visual impairment globally are uncorrected refractive errors, cataracts, and glaucoma. Refractive errors include near-sightedness, far-sightedness, presbyopia, and astigmatism. Cataracts are the most common cause of blindness.

With Classification:

- 20/200 to 20/400 : is considered severe visual impairment, or severe low vision
- 20/500 to 20/1,000 : is considered profound visual impairment, or profound low vision
- More than 20/1,000 : is considered near-total visual impairment, or near total blindness
- No light perception (NLP) : is considered total visual impairment, or total blindness



Management for Mobility:

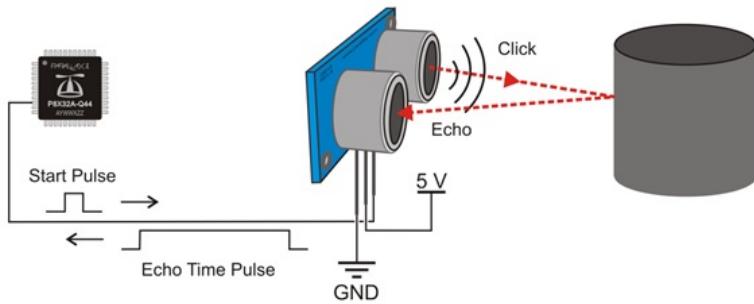
- Getting training from orientation and mobility specialists professionals who are specifically trained to teach people with visual impairments how to travel safely.
- Tools such as the white cane with a red tip – the international symbol of blindness – may also be used to improve mobility.
- A small number of people employ guide dogs to assist in mobility.
- GPS devices can also be used as a mobility aid. Such software can assist blind people with orientation and navigation.
- Some blind people are skilled at echolocating silent objects simply by producing mouth clicks and listening to the returning echoes

Source: Wikipedia

Theory

Ultrasonic sensor

The PING))) sensor requires a brief high/low signal called a start pulse from the Propeller chip to start the measurement. The sensor then makes its ultrasonic click and sets its SIG pin high. When the echo comes back, it sets its SIG pin low. The Propeller chip then can measure the duration of the PING))) sensor SIG pin's high signal, which corresponds to the echo return time. We have used the ping library available for this sensor in Propeller C.



Wireless Earphones

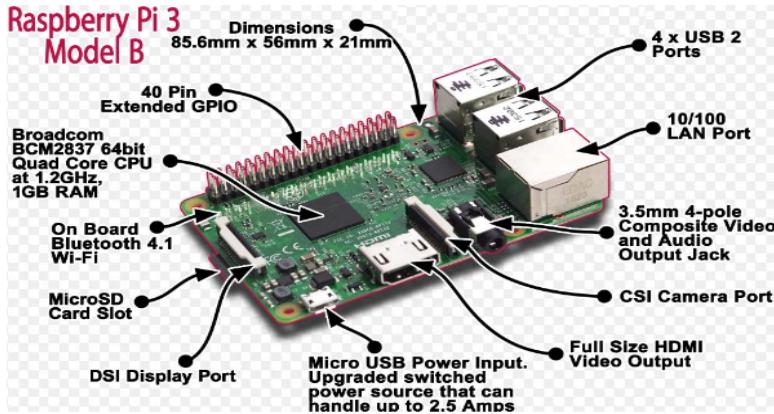
Wireless earphones are speakers which receive audio signals using radio frequency (RF) waves. The two most popular RF frequencies that support audio transmission to wireless speakers include a variation of WiFi with frequency range used by cordless telephone of 900 MHz, while others depend on Bluetooth to transmit audio data to the receiving speaker. Wireless speakers are composed of two units: a main speaker unit combining the speaker itself with an RF receiver, and an RF transmitter unit. The transmitter connects to the audio output of any audio devices such as hi-fi equipment, televisions, computers, mp3 players, etc.



Raspberry Pi

The Raspberry Pi is a small single-board computer developed in the United Kingdom by the Raspberry Pi Foundation. A ready to use out of credit card sized Linux Computer which can be mounted on robots or any other mobile divide which needs on board computational power, and can be interfaced with other microcontrollers using low level peripheral. Uses RISC architecture and is based on BCM2837 ARM processor with specifications

Raspberry Pi 3



Based On ARMv8 with 1.2GHz 64-bit quad-core has 1GB RAM. With on board WiFi and Bluetooth, 40 GPIO pins, 4 USB2.0 ports with up to 1.2A output, full HDMI port, ethernet port, combined 3.5mm audio jack and composite video, camera interface (CSI), display serial interface (DSI), Micro SD card slot, VideoCore IV 3D graphics core and on board voltage regulator.

Arduino

Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. Arduino is a tool for making smart devices that can sense and control the physical world. Arduino microcontroller is programmed using Arduino programming language.



Arduino Uno

is ATmega328 processor, 16Mhz clock speed with features:

- port system (6 analog inputs, 14 digital I/O (6 support PWM output))
- 3 timer system (two 8 bit and one 16 bit)
- analog-to-digital converter (ADC) (6 channels with 10 bit resolution)
- interrupt system (26 interrupts : 24 internal and 2 external)
- serial communications (SPI, I2C and USART)
- Memory system (32 KB ISP, 2KB EEPROM and 1 KB addressable EEPROM)

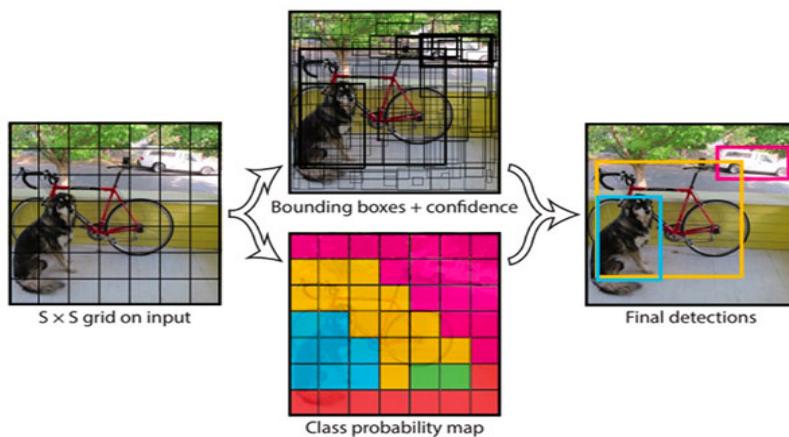
USART

A USART — a Universal Synchronous/Asynchronous Receiver/Transmitter — is a microcontroller peripheral that converts incoming and outgoing bytes of data into a serial bit stream. Synchronous operation uses a clock and data line while there is no separate clock accompanying the data for Asynchronous transmission. There are several registers used to control the USART.

Yolo Object Detection

Yolo detects objects in both images and video streams using Deep Learning, OpenCV, and Python. Methodology for object detection generally falls into either machine learning-based approaches or deep learning-based approaches. It deals with detecting instances of semantic objects of a certain class (such as humans, buildings, or cars) in digital images and video. Every object class has its own special features that helps in classifying the class – for example all circles are round. Object class detection uses these special features.

What is the YOLO object detector?



source:<https://www.pyimagesearch.com/2018/11/12/yolo-object-detection-with-opencv/>

Finger Tracking

Finger detection is an important feature of many computer vision applications. In this application, A histogram based approach is used to separate out the hand from the background frame. Thresholding and Filtering techniques are used for background cancellation to obtain optimum results.

Process:

- Skin color histograms are used. This histogram is then used to subtracts the background from an image, only leaving parts of the image that contain skin tone.
- Rectangles are drawn on the frame and the user places their hand inside these rectangles. We take skin color samples from the user's hand and then create a histogram.
- We take the pixel values from rectangles and put them in a matrix and form a skin histogram.
- Changed the input frame to HSV and then applied cv2.calcBackProject with the skin color histogram and used it to find the components of the frame that contains skin and used a Filtering and Thresholding function to smoothen the image.

- Using OpenCV found contours in a frame used convexity defects, which are potential fingertip locations which are furthest from the centroid of the contour.
- Created a list to store the changed location of the farthest_point in the frame.

source: <https://dev.to/amarlearning/finger-detection-and-tracking-using-opencv-and-python-586m>

Pose Estimation

In order to determine the distance from our camera to a known object or marker, we utilized triangle similarity.

Process:

Let's say we have a marker or object with a known width W . We then place this marker some distance D from our camera. We take a picture of our object using our camera and then measure the apparent width in pixels P . This allows us to derive the perceived focal length F of our camera:

$$F = (P \times D) / W$$

As we continue to move my camera both closer and farther away from the object/marker, we applied the triangle similarity to determine the distance of the object to the camera:

$$D' = (W \times F) / P$$

Source: <https://www.pyimagesearch.com/2015/01/19/find-distance-camera-objectmarker-using-python-opencv/>

Process

The entire working process for the Cameye is complex yet very easy to use. The person who is using the device just needs to use his phone camera and wear a belt around his waist. Modes of operation:

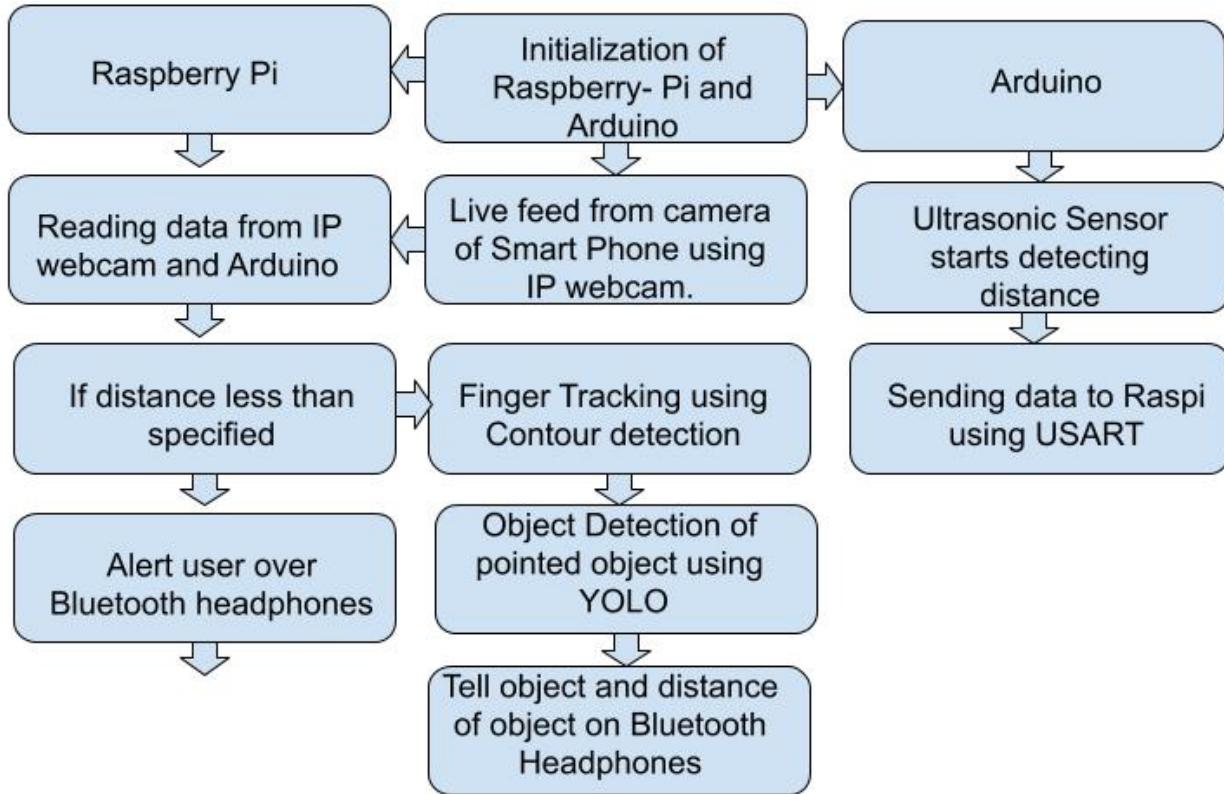
- Camera mode for object identification.
- Proximity mode for avoiding collision.

Camera Mode:

Phone camera is connected to Raspberry Pi using PI webcam using this it is converted into IP cam, streaming is achieved through WiFi using URLLib and OpenCvstreams live feed is processed on Raspi using Yolo for object detection where the processing is done using computer vision algorithm and OpenCv to find contour to located the finger. If a person wants to know what's in front of him he can point his finger towards the object .

Proximity Mode:

A ultrasonic sensor is placed on the user which is interfaced with arduino as soon as a object is within a given proximity range it notifies the user about the object and helps him from colliding with the object the arduino sends data to raspi using USART where it is processed to notify the user.



Future work

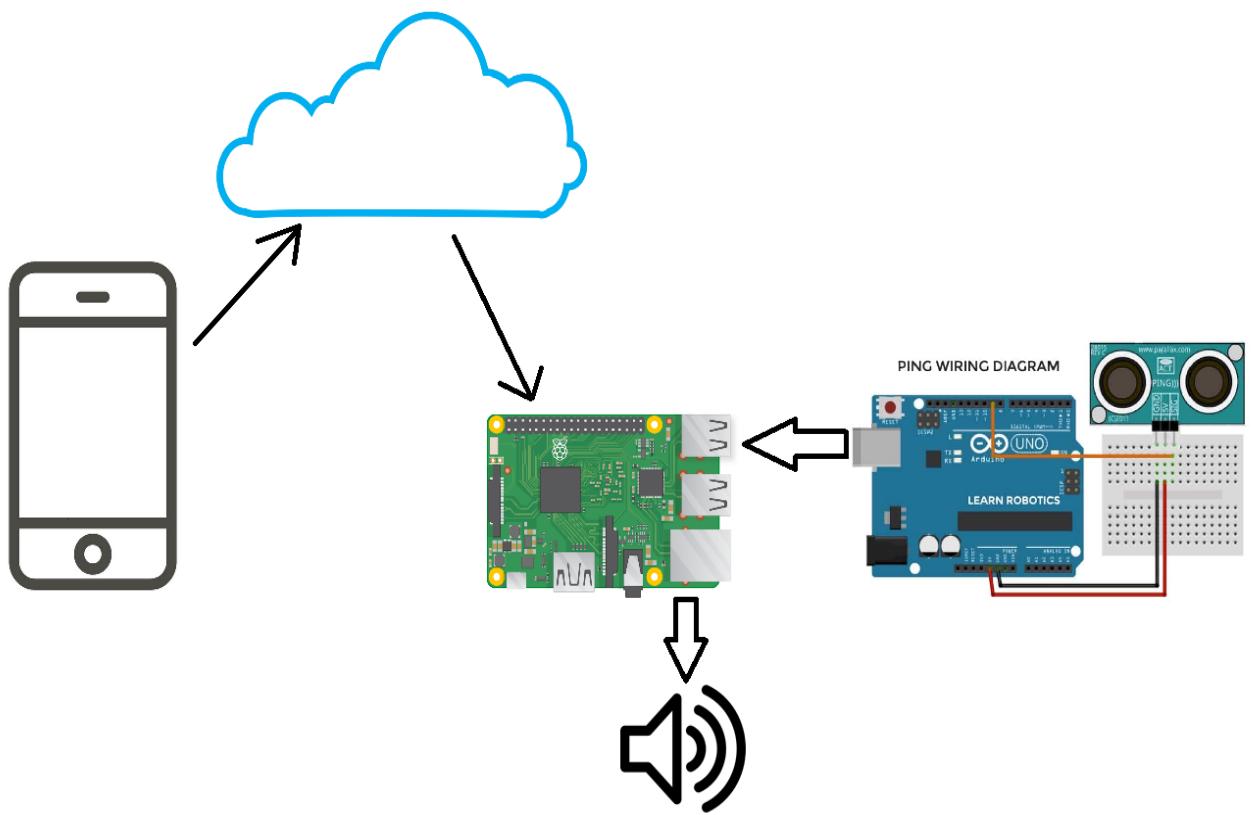
- Making a belt with microfiber actuators.
- Using Epipolar geometry to find the distance of an object.
- Integrating with another Board having faster processing power

Pose Estimation (Future Work)

Process

Finding depth using my phone camera and accelerometer, initializing phone at a starting point and call it as the world frame, then moving camera to a new point say p2 accelerometer gives me rotation and translation which is noisy, using 8 point algorithm for the two images taken from the two poses one can find out the essential matrix which gives me rotation and translation up to a scale, using this rotation from essential matrix find out the angle of rotation .Make an update to accelerometer data using Kalman Filter and then use this data to find the depth of my object and keep changing my reference frame as I move forward.

Circuit Diagram



Bill of Materials

Items	Quantity	Price
Raspberry Pi model 3B+ With Kit	1	\$65
Arduino Uno	1	\$13

Ultrasonic sensor	1	\$5
TOTAL		\$83