Developing a Navigation System Utilizing Stereoscopic Cameras for the Visually Impaired

Ankit Yande

Background Info

- An estimated 253 million people live with vision impairment, 36 million of whom are blind (WHO 2017)
- There are many ways of calculating distance however many of them are invasive such as ultrasonic sensors, laser distance sensors, infrared sensors, etc.
- Humans have depth perception because they have two eyes
  - Our brain uses the difference of vision between our two eyes to estimate distance
- This behavior can be replicated using stereo vision (also known as binocular vision, or stereopsis)
- In 1838, English physicist Charles Wheatstone described the theory of stereoscopic vision and created his mirror stereoscopic viewer
- Using the offset between two sensors, the distance between a stereoscopic camera and an object in the frame can be calculated

Goal

To develop software in Java using OpenCV libraries that can successfully detect an object in two different feed using characteristics of the object such as shape, hue, and brightness and then calculate the distance between that object and the cameras using trigonometry. This information should then be related to a human in an audio format.

Methods

Phase 1) Image Detection

- An image detection algorithm was created in Java using OpenCV (Open Source Computer Vision) libraries
- A tennis ball was chosen to be the detected object
- A maximum and minimum value for the hue, saturation, and value (HSV) of a tennis ball were determined experimentally and these values were used to create a mask of areas in the image that was the within the correct color range. Noise was reduced by eroding the mask and clarity was added after by dilating the mask
- A bounding box was created using the boundaries of the region that remained and the center was calculated

Phase 2) Distance Calculations

- Trigonometry calculations were then performed by the computer using the center of the object on both feeds, the distance between cameras, the field of view of the cameras, and the resolution of the cameras

Phase 3) Output

- Using the distance data calculated in Phase 2, the computer will give audio directions to the position of the object such as "The object 5 meters away"

```
static Scalar min = new Scalar(30, 100, 60);// HSV
static Scalar max = new Scalar(60, 255, 255);// HSV
    Mat imgThreshold = new Mat();

    Imgproc.cvtColor(inputframe, imgThreshold, Imgproc.COLOR_BGR2HSV);

    Core.inRange(imgThreshold, min, max, imgThreshold);
```

Figure 1) The maximum and minimum HSV values were determined of the object that was to be detected. A threshold matrix was created that served as a mask of the pixels in the correct range.

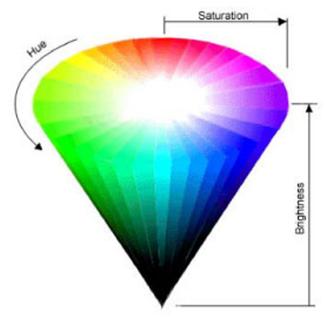

Figure 2) A visual representation of Hue Saturation and Brightness (Jewett, www.tomjewett.com).

```
Mat kernelOpen = Mat.ones(15, 20, 0);
Mat kernelClose = Mat.ones(50, 50, 0);
Imgproc.morphologyEx(imgThreshold, imgThreshold, Imgproc.MORPH_OPEN, kernelOpen);
Imgproc.morphologyEx(imgThreshold, imgThreshold, Imgproc.MORPH_CLOSE, kernelClose);
```

Figure 3) Matrices were created that were used to erode and dilate the mask.

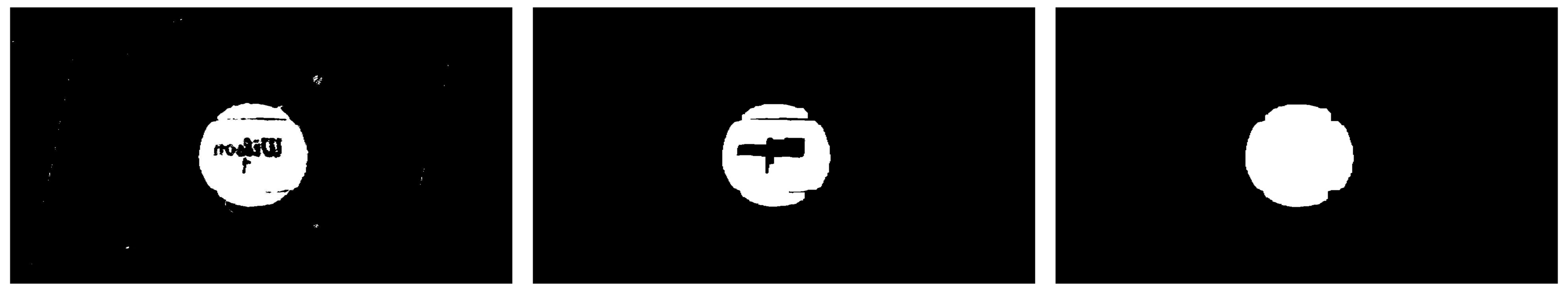

Figure 4) Visual representation of eroding and dilating the mask. The first image is the mask straight out the camera. The second image is after eroding the mask to get rid of noise. The third image after re-dilating the image to fill the gaps back in.

```
List<MatOfPoint> contours = new ArrayList<MatOfPoint>();
Imgproc.findContours(imgThreshold, contours, mat1, Imgproc.RETR_EXTERNAL, Imgproc.CHAIN_APPROX_NONE);

for (int i = 0; i < contours.size(); i++) {

    Rect rect = Imgproc.boundingRect(contours.get(i));
    if (Math.abs((double)rect.width/(double)rect.height)<1.5 &&
        Math.abs((double)rect.height/(double)rect.width)<1.5) {

    Imgproc.rectangle(inputframe, rect.tl(), rect.br(), new Scalar(0, 255, 0), 2);
    center = new Point((rect.tl().x + rect.br().x) / 2, (rect.tl().y + rect.br().y) / 2);
    Imgproc.rectangle(inputframe, center,center,new Scalar(255, 0, 0), 10);

     }
}
```

Figure 5) Using OpenCV's findContours method, an Array of points were created that outlined the white areas of the mask. A bounding rectangle was then created using the values of the contour. An if condition was used to only create a bounding box if the width and height of the box were similar due to the spherical nature of the object. Finally, the center point of the rectangles was found to be used in calculations.

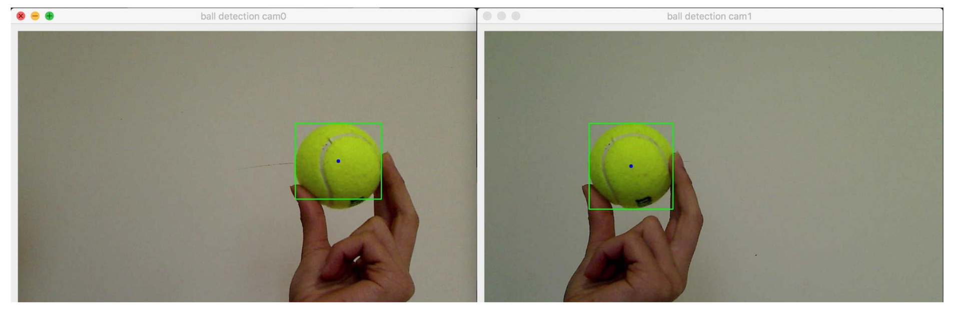

Figure 6) The output of the image recognition process with the bounding rectangle and center point displayed.

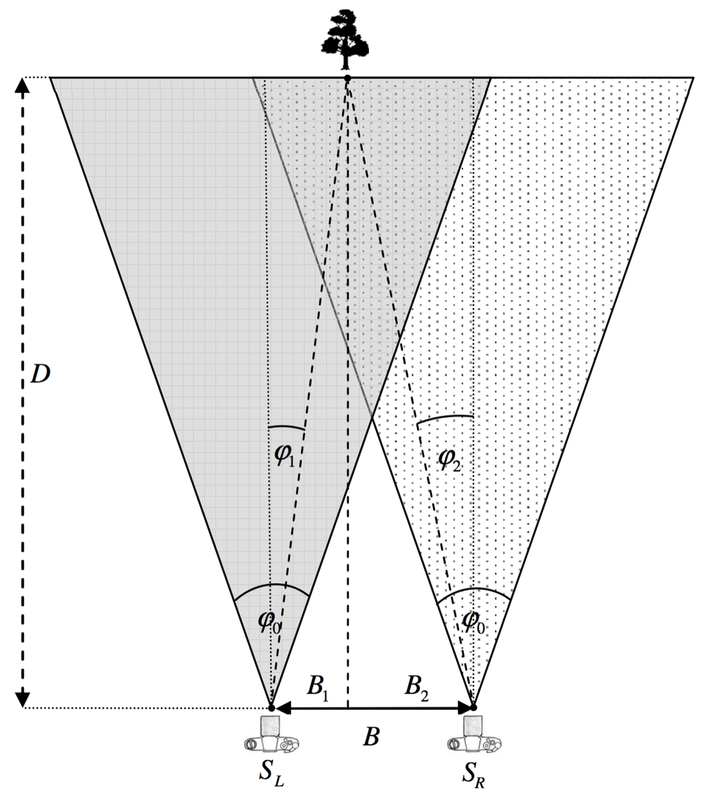

Figure 7) Diagram of how the the stereo cameras were set up. Based on the diagram the following equation was derived to calculate distance:

$$D = \frac{Bx_0}{2(\Delta x)(tan\frac{\phi_0}{2})}$$

(Mrovlje and Vrančić 2008).

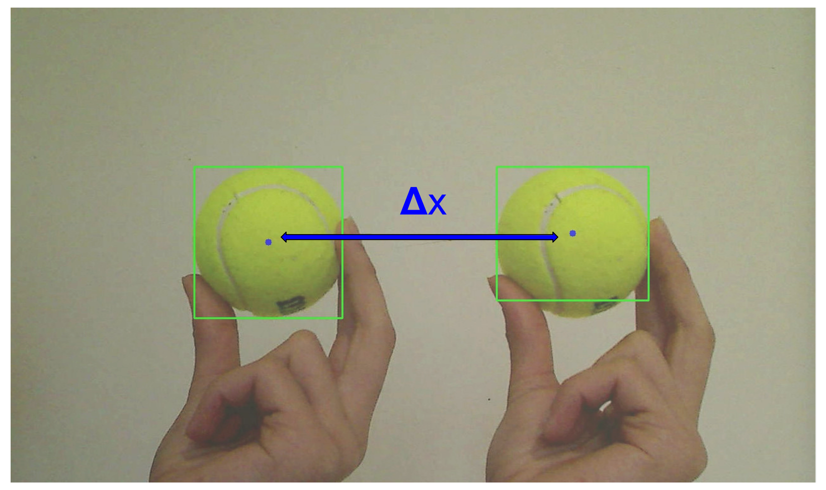

Figure 8) The image from the left and right cameras overlayed. Disparity (Δx) is the distance in pixels between the centers (indicated with blue dots) of the object in each frame.

Figure 9) A tennis ball was chosen as a test object and disparity readings were taken at 10 cm intervals. This data was then used to create a distortion constant k to create better estimations of distance using the equation:

$$D = \frac{Bx_0}{2(tan\frac{\phi_0}{2})(\Delta x)^k}$$

Tests were then done in bright and low light environments to see how accurate the computer calculated distances were.

Results

As expected, there was an inverse correlation between distance and disparity. The data from the trials was used to create a constant to account for the distortion of the cameras. The code worked as expected an was able to detect an object and give an accurate reading of distance

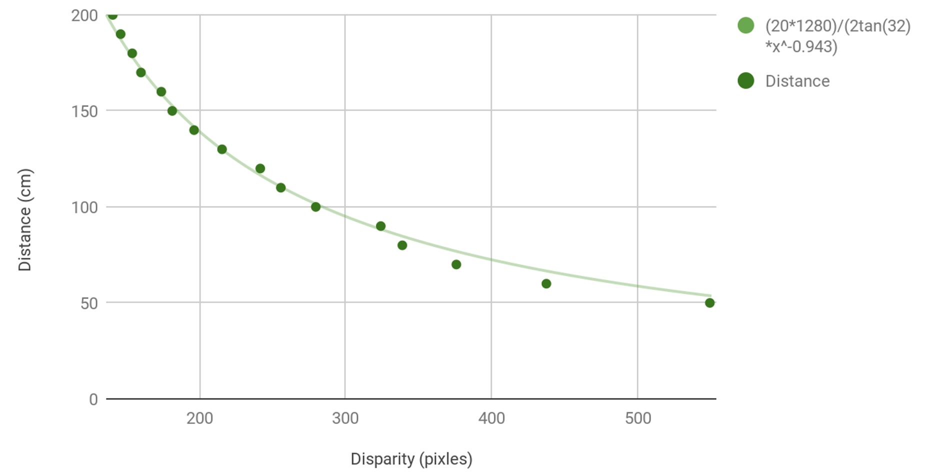

Figure 10) Depicts the inverse correlation between the distance the object is from the camera and the disparity (Δx) in pixels between the center of the object in each frame. A best fit line was determined by using the trigonometric formula calculated and adjusted with the coefficient k that was experimentally determined

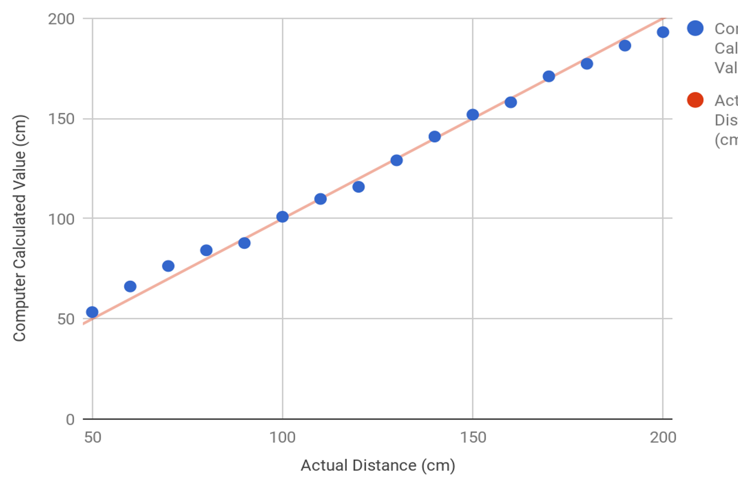

Figure 11) The computer calculated value compared to the actual distance taken in a well lit environment

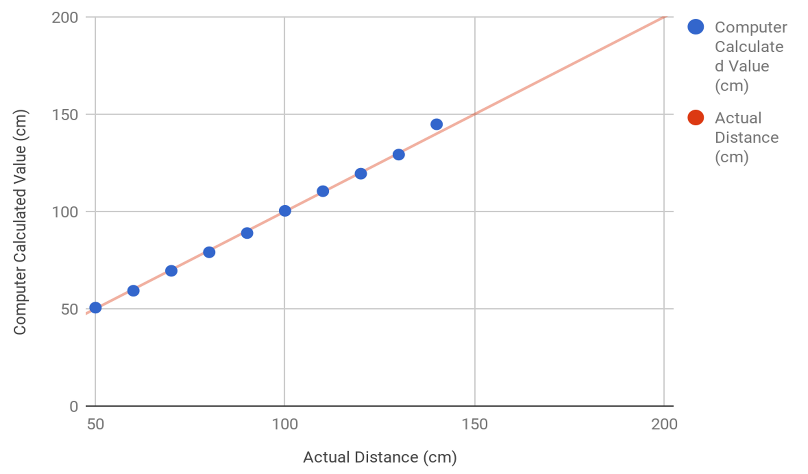

Figure 12) The computer calculated value compared to the actual distance taken in a dimly lit environment. After 150cm, the cameras could not properly recognize the test object

Discussion

A program was successfully developed that could find an object in two camera frames and calculate distance accurately. This distance was successfully outputted via a text to speech program. Tests were done in which a tennis ball was moved in 10 cm intervals and a computer reading of the distance was taken. The computer gave accurate measurements in both lighting conditions however the ball was not able to be recognized after it was 150cm from the cameras.

Limitations

- Maximum Resolution of the Webcams used
- Color variation between the two cameras

Possible application

- This project could be used to help the visually impaired be better aware of and locate objects in their surroundings

Future experiments

- Better calibration and color matching of the cameras
- Calculating the maximum and minimum HSV values based on the averages of the image
- Creating a mobile app for a phone with dual cameras
- Working with people with visual impairments to determine the best possible user feedback
- Developing more sophisticated detection algorithms or training a neural network to detect the object

Literature Cited

"03-Continually Detect Face in Webcam - Pdopencvjava." *Pdopencvjava*, sites.google.com/site/pdopencvjava/webcam/02-continually-detect-face-in-webcam.

Anirban. "OpenCV Object Tracking by Colour Detection in Python." *The Codacus*, 13 Aug. 2017, thecodacus.com/opencv-object-tracking-colour-detection-python/#.WqBu75PwbfY.

"FreeTTS 1.2.3 ." *FreeTTS 1.2.3 - A Speech Synthesizer Written Entirely in the Java(TM) Programming Language*, Sun Microsystems, Inc, freetts.sourceforge.io/#what_is_freetts.

Jewett, Tom. "HSB: Hue, Saturation, Brightness." *Color Tutorial*, www.tomjewett.com/colors/hsb.html.

Mrovlje1, Jernej, and Damir Vrančić. "Distance Measuring Based on Stereoscopic Pictures ." dsc.ijs.si/files/papers/s101%20mrovlje.pdf.

"OpenCV Libraries." *OpenCV*, OpenCV Team, opencv.org/about.html.

"Vision Impairment and Blindness." *World Health Organization*, World Health Organization, Oct. 2017, www.who.int/mediacentre/factsheets/fs282/en/.

Acknowledgements
The author would like to thank his teacher and supervisor, Mr. Danch, for his support.