Application of Computer Vision Based Gaze and Body Tracking

Anushiya Balakrishnan, Sean Wu, Samuel Yang & Justen Yeung
{ab112, sww35, sy369, jy388}@scarletmail.rutgers.edu
Advisor: Prof. Kristin Dana

RUTGERS
THE STATE UNIVERSITY OF NEW JERSEY

Goal

Design an affordable camera and computing module to achieve both of the following tasks:

- Track the head poses of multiple users in order to extrapolate the location of their gaze
- Track pedestrians as they walk through a scene, while accounting for occlusion

Motivations and Objectives

The market for data analytics has grown recently as data collection and data visualization tools improve and enter the mainstream, but implementations are yet to exist for physical attractions such as museums and national parks.The aim of this project is to create a platform for vision based software that analyzes the effectiveness of attractions like advertisements and museum exhibits.

- ❑ Facial landmark detection
- ❑ Head pose estimation
- ❑ Pedestrian tracking
- ❑ Position extrapolation

Research Challenges

- ❑ Real time gaze tracking without user input
- ❑ Optimizing and increasing performance of vision software
- ❑ Eliminating false positive pedestrian readings
- ❑ Camera property adjustment for background subtraction
- ❑ Gaze detection without specialized hardware
- ❑ Inaccuracies in position extrapolation

Acknowledgement

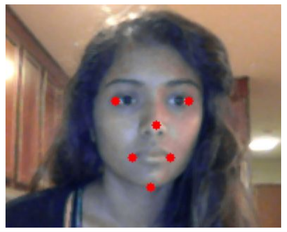
We would like to thank Professor Dana for her guidance and feedback on this project and Professor Godrich for overseeing the Capstone Program.

Methodology

A consumer-grade webcam is used for video input. Vision is coded largely in Python, using OpenCV and dlib libraries.

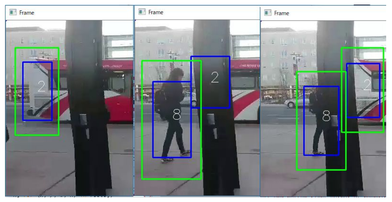
❑ Gaze Tracking

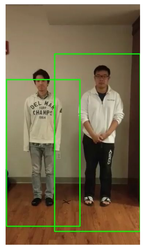
- Six facial landmarks are detected and compared to an approximated 3D model of an average face
- The difference between the position of detected landmarks and model landmarks is computed using the PnP method to obtain orientation of the head



❑ Pedestrian Tracking

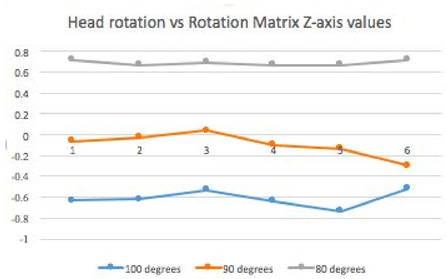
- Detection using cascade classifier with histogram of gradients
- Detection using background subtraction as a foundation for blob tracking
- Position of occluded pedestrians are linearly extrapolated based on velocity





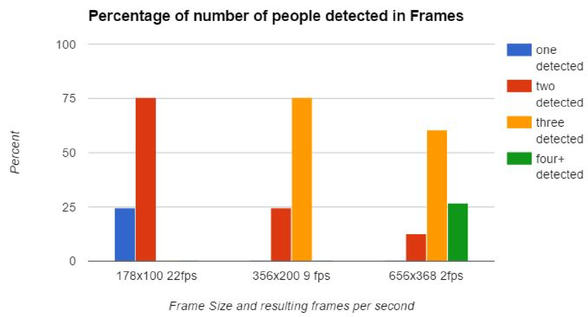
Results

- ❑ Head pose can be extracted (approximately) from rotation matrix output by solvePnP()
- ❑ Currently accurate in three large directions - can identify gaze within a large 3x3 grid
- ❑ Future work is to refine accuracy by reducing noise - current landmarking is noisy and slow; errors with correct zone identification




```
Rotation Vector:
 [[ 0.00480995]
 [-3.90906298]
 [-0.46703503]]
left
Rotation Vector:
 [[-2.94888679]
 [-0.14982782]
 [ 1.24865434]]
right
Rotation Vector:
 [[-3.01761041]
 [-0.04927534]
 [ 1.15347499]]
right
Rotation Vector:
 [[-3.00088257]
 [-0.06609513]
 [ 1.10201176]]
right
Rotation Vector:
 [[ 2.90223457]
 [ 0.04371667]
 [-1.06922621]]
```

- ❑ Determined optimal operating parameters for HOG
- ❑ In the images below, the anticipated location of bounding box 2 is demonstrated in the 3rd frame, despite the occlusion in the video
- ❑ Future work is to continue exploring pedestrian detection methods to increase accuracy and computational efficiency



References

[1] OpenCV Documentation http://docs.opencv.org/2.4/
[2] Head Pose Estimation http://www.learnopencv.com/head-pose-estimation-using-opencv-and-dlib/
[3] HOG Algorithm http://www.pyimagesearch.com/2015/11/16/hog-detectmultiscale-parameters-explained/