A BRIEF REPORT OF THE SOLUTION – A PROTOTYPE FOR HUMAN AVOIDANCE.

- Created by Amandeep Jiddewar

Introduction:

The project aims to create a prototype of a component of a robotic depth perception system for human avoidance. The task is to find a distance between a human and a wall in a corridor using depth camera observations.

The purpose of the perception system is to ensure the robots avoids a human in a corridor. Given an image from a depth camera, the perception system should output two values:

1. Which side is safer to avoid a human ("left" or "right")?
2. What is the clearance for avoidance maneuver? Clearance is defined as the smallest distance between a human and the obstacle closest to him (a wall or a shelf).

Assumptions Made:

Below are the assumptions made:

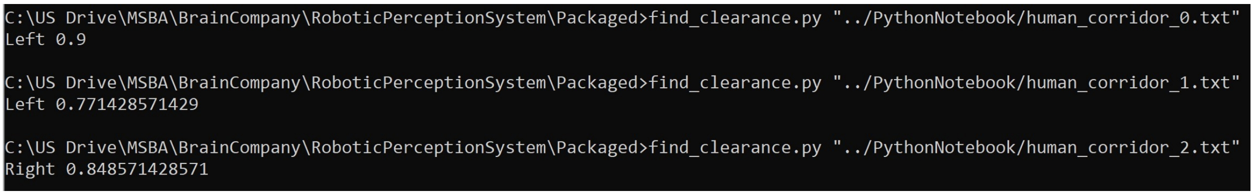
1. Only one human is present in the corridor.
2. The noise threshold in depth image is a hard-coded value of '5'.
3. The human is always present in the corridor.
4. The human can be successfully distinguished and is always between the shelf and the wall.
5. The distance calculated is from the center (along the x-axis) of the rectangle that bounds the human.
6. The distance of the human is 2 meters.
7. The width of the corridor is 1.5 meters.
8. The camera is always approximately at the center of the corridor.

Possible next steps:

1. Make the solution work with multiple humans present in the corridor.
2. The human detection can be done faster by using faster detection implementations such as YOLO.
3. The noise removal part should be generalized to accommodate any surrounding.
4. The system should be generalized for robot present near to the shelf or near to the wall. (robot anywhere in corridor width-wise)

5. The system can be extended to detect other objects apart from humans.
6. The system may leverage data from the location services to be able to give precise distances between objects and plan a route in accordingly.
7. The system should be able to find a path in a video stream or real-time and when human and/or robot are in motion.
8. Other advanced next steps would be to create a dynamic map to avoid obstacles, much like the ones in self-driving cars.

Sample solution screenshot:



```
C:\US Drive\MSBA\BrainCompany\RoboticPerceptionSystem\Packaged>find_clearance.py "../PythonNotebook/human_corridor_0.txt"
Left 0.9

C:\US Drive\MSBA\BrainCompany\RoboticPerceptionSystem\Packaged>find_clearance.py "../PythonNotebook/human_corridor_1.txt"
Left 0.771428571429

C:\US Drive\MSBA\BrainCompany\RoboticPerceptionSystem\Packaged>find_clearance.py "../PythonNotebook/human_corridor_2.txt"
Right 0.848571428571
```

More Notes:

1. GitHub repo: https://github.com/amandeepfj/RoboticPerceptionSystem
2. The solution is tested on both Python 2.7 and 3.
3. The requirements are mentioned in the requirements.txt file inside the "Packaged" folder in the repository.

References:

1. https://funvision.blogspot.com/2017/01/lbp-cascade-for-head-and-people.html
2. https://opencv.org/
3. https://www.pyimagesearch.com/2015/01/19/find-distance-camera-objectmarker-using-python-opencv/
4. https://docs.opencv.org/2.4.13.7/doc/tutorials/objdetect/cascade_classifier/cascade_classifier.html
5. https://www.pyimagesearch.com/start-here-learn-computer-vision-opencv/