**Midterm Project - Phase 0 Proposal: Group 8**

* **Abstract:**
* For the midterm project, me and my teammate have decided to choose **Acme’s perception robotic module** & work on the deliverables and provide a high-quality solution following the essential software engineering practices which they can directly integrate into their product.
* We have decided to deploy the solution in a **hardware-based model**. For that we will be using either Laptop with its integrated web camera or Raspberry Pi with a dedicated monocular video camera.
* **Deep learning and Open CV** (pre-trained model) will be deployed on Linux, which will be acting as the **Human Detector** and **Human Tracker.** The python & C++ script will be modified according to our requirement.
* After the modifications, the program or module will output spherical co-ordinate of the humans with respect to robot frame along with each human’s unique ID.
* **Flow Chart:**

Graphical user interface

Description automatically generatedA screenshot of a phone

Description automatically generated with medium confidence

**Fig 2: Detailed Flow Representation**

**Fig 1: Top Level Diagram**

**Fig 2: Detail Diagram**

* **Assumptions:**

1. Floor is flat.
2. Human height is approximately 5.5 ft.
3. Camera on robot is perpendicular to the floor (In our case, the laptop webcam is perpendicular to the floor)
4. No distortion from camera lens. It is assumed that we use pin-hole camera.

* **Human Detector:**

1. Get an “Image Frame” from camera.
2. Detect human using the pre-trained model.
3. Get pixel value (a,b) of the head of the human.
4. Calculate spherical co-ordinates of the camera lens (r1, theta1, phi1) by assuming robot location/coordinate as (0,0,0). The height of camera lens from floor (or in other words location of camera lens in 3d space).
5. Calculate the radial distance ‘r’ of the human using the camera FoV, total pixel count, camera lens coordinate and basic math.
6. Convert (a,b) [cartesian co-ordinate] to (r, theta, phi) [spherical co-ordinate], where ‘r’ is found in the previous step.
7. Convert the coordinates from camera frame to robot frame, i.e.., (r, theta, phi) to (r2, theta2, phi2) 🡪 loc of human in 3D space w.r.t robot frame.

* **Human Tracker:**

1. Assign IDs for detected human after step (2) of “Human Detector”. Also get (a,b) pixel coordinate of this human’s head. Let this ID be “Human 1”.
2. Get next I/P frame from “Human Detector” & also pixel coordinate (a1,b1) of detected human’s head.
3. Compare previous frame human pixel coordinate (a,b) with current frame human pixel coordinate (a1,b1) & if it is greater than a certain threshold, assign a new ID (“Human 2”).

(If the distance between human 1 and 2 is above certain threshold, it is considered as a new human & given new ID – this threshold is some min pixel distance since we dedicatedly do not handle occlusion)

1. Repeat the process for consecutive frames using the python & C++ algorithm and assign unique IDs for multiple humans detected in a frame.