

Person Detection & Waypoint Algorithm

The person detection algorithm we implemented identifies people in the camera's FOV and tracks them (using a Euclidean distance heuristic) frame-to-frame. It considers targets "locked" if detected in the FOV for 30 consecutive frames. Upon identifying a set of locked targets, the world coordinates of the people get turned into waypoints for the robot to travel to along a multi-segment trajectory.

For each frame:

- Get depth directly from RGB-D camera stream
- Run MobileNet image classifier:
 - Filter to person objects with confidence > 0.7
 - Capture bounding box around any persons identified
- If person(s) detected in frame:
 - Merge depth map and bounding boxes
 - Compute bounding box x, y, z world coordinates using depth and camera intrinsics
 - Check if the person matches a previously identified target
 - Is the person (in world coordinates) within 0.2m a previously identified target?
 - If yes, increment the previously identified person's match count
 - Recompute the previously identified person's place in the world using a recursive mean
 - If no, add the person to tracked persons
- For all identified persons
 - If all identified persons > 30 match count: Targets Locked
 - If some identified person < 30: Targets not locked

Motion Control Algorithm

We implemented a trapezoidal trajectory for motion control using maximum acceleration and velocity values to ensure smooth motion and to avoid spilling any payload contents.

```
float decel_disp = fabs(target_disp) - (prev_vel * prev_vel) / (2 * max_acc);
if (fabs(curr_disp) < decel_disp) {
    target_vel = prev_vel + dir * max_acc * dt_s;
    target_vel = Saturate(target_vel, max_vel);
} else if (fabs(prev_vel) > max_acc * dt_s) {
    target_vel = prev_vel - dir * max_acc * dt_s;
} else {
    target_vel = 0;
}
```

- P_new =
- V_max
- A_max