EC4098D-Project Part-2

Autonomous Robots for Transportation of Inventory in a Warehouse

Group 7 :

- Adithya Shaji - B170218EC
- Anandhu C C - B170292EC
- Asna P - B170311EC
- Athul P - B170509EC

Guide : Dr. Praveen Sankaran

Outline

1. Phases of Project
2. Progress After first presentation
3. Pure Pursuit Controller
4. Tracking the path
5. Sample path tracking video
6. Object detection
7. Conclusion
8. Reference

Objective

Warehouses are filled with repeatable, process-oriented, and error-prone operations. With automation, robots can take over repetitive tasks like unloading and transporting packages from humans to make the process smoother, faster and efficient.

- Modeling the environment of a warehouse and deploying robots in it.
- A series of task to load and transport objects from one location to its destination is allocated to a series of robots.
- Robot will use pathfinding algorithm to find the shortest distance to the destination.
- Using Path tracking algorithm the robot track its path.
- It will detect the obstacles on its way by using computer vision algorithms and choose an action that will help it avoid collision.
- The robot will unload the object once it reaches its destination

Phases of Project

1. Familiarization and test environment modelling ✓
2. Path planning ✓
3. Path tracking ✓
4. Object detection and avoidance ✓
5. Pick and drop
6. Introduction of multiple robots
7. Scheduling and task allocation
8. Development of user interface

Path Planning

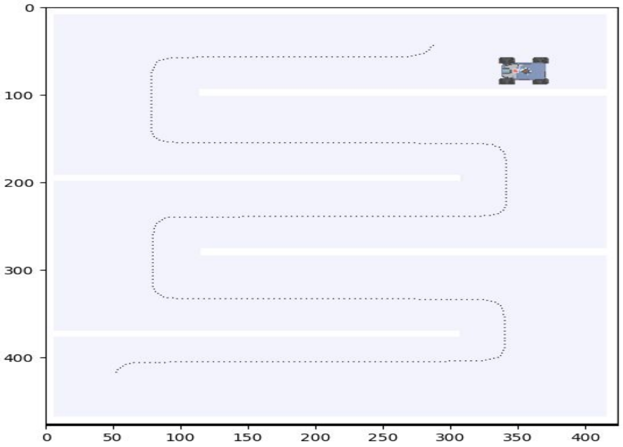
- In the previous part we have done Path planning using A* algorithm ,robot should fed with a path from initial position to its destination.
- The robot then tracks the calculated path using a path tracking algorithm -pure pursuit controller.

Progress after first Presentation

Path Smoothing

- The path from the A* algorithm should be smoothed to avoid sharp turns.
- The algorithm takes in a 2D array of XY coordinates and returns a smoother version of the coordinates.
- It takes in three parameters: weight_data which we call
- (a)weight_smooth (b) tolerance . The amount this algorithm smooths depends on the spacing between the points and the values of a , b , and tolerance .
- A larger b means a smoother path.

Smoothed Path



Pure Pursuit Controller

- Pure pursuit is a path tracking algorithm.
- It directs the robot to travel along an arc from the current location to the goal point. This goal point is called the lookahead point and it is a point on the path that is the lookahead distance from the robot.
- As the robot moves along the path,it aims for the lookahead point, which moves down the path with the robot.
- The robot can be considered to “pursue” the lookahead point. This action is analogous to how a human driver looks at a point down the road and aims towards that point.

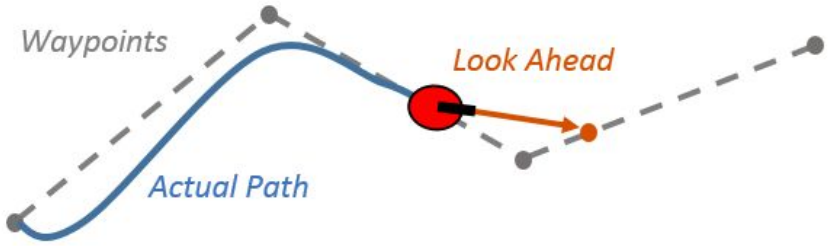
Tracking the path

The algorithm to follow the path is as follows:

- Find the lookahead point from the point which is nearest to the robot
- Calculate the curvature of the arc to the lookahead point
- Calculate the corresponding left and right wheel velocities to achieve that curvature.
- Update the target left and right wheel velocities of the robot in the simulator.

Lookahead point

The lookahead point is the point on the path that is the lookahead distance from the robot. We find the lookahead point by finding the intersection point of the circle of radius lookahead distance centered at the robot's location, and the path segments.



Path Tracking Demo

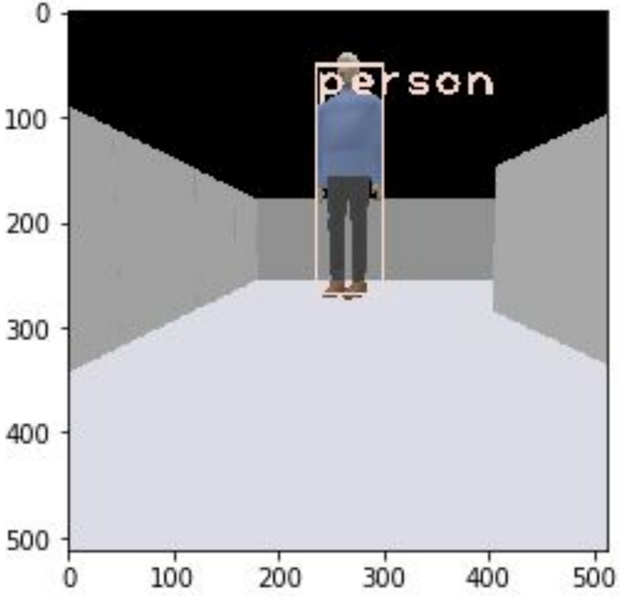
Object Detection

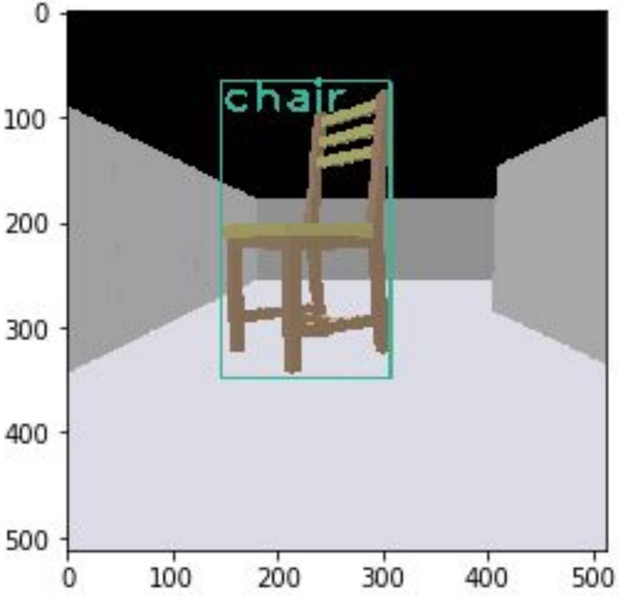
- A vision sensor is placed on the robot. Distance of range of the vision sensor can be varied.
- Once vision sensor detects an object the robot stops moving and the image captured is passed onto the object detection system.
- The object detection system works on YOLOV3 algorithm which is neural network object detection algorithm ,where it segments and correctly identifies objects with just a single image.
- The object is identified.

YoloV3

- Prior detection systems repurpose classifiers or localizers to perform detection. YoloV3 use a totally different approach. It apply a single neural network to the full image.
- This network divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities.
- YoloV3 makes predictions with a single network evaluation unlike systems like R-CNN which require thousands for a single image.
- This makes it extremely fast, more than 1000x faster than R-CNN and 100x faster than Fast R-CNN

Object Detection of sample objects





Conclusion

The path planning algorithm and object detection was implemented successfully. We were able to identify and move the robot in a correct path. Obstacles were correctly detected and identified.

References

- https://pysource.com/2019/06/27/yolo-object-detection-using-opencv-with-python/
- H. Gong, H. Li, K. Xu and Y. Zhang, "Object Detection Based on Improved YOLOv3-tiny," 2019 Chinese Automation Congress (CAC), Hangzhou, China, 2019, pp. 3240-3245, doi: 10.1109/CAC48633.2019.8996750.
- Coulter, R. Craig. Implementation of the pure pursuit path tracking algorithm. Carnegie-Mellon UNIV Pittsburgh PA Robotics INST, 1992.
- https://www.chiefdelphi.com/t/paper-implementation-of-the-adaptive-pure-pursuit-controller/166552
- Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement arXiv
-