Autonomous Robots for Transportation of Inventory in a Warehouse

Final Presentation-Group 7

Athul P Asna P Anandhu C C Adithya Shaji

April 19, 2021

National Institute of Technology Calicut

Introduction

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.

Phases of Project

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

Work done in S7

Work done in S7

- CoppeliaSim was identified as the best simulation software for the project, and a test environment was designed to familiarize with the software.
- Path Planning: We used A* algorithm to find path from initial position of pickup point and from pickup point to the drop point.
- Path Tracking: We employed Pure Pursuit controller algorithm for the robot to track the path found from the previous step.
- Object Detection: We were able to detect objects via the camera mounted in-front of the robots via YOLOv3 model.

Work done in S8

Object detection

- The object detection feature was dropped because it slowed the whole simulation down.
- We assumed that all of the objects in simulation are static except the robots.
- The static object avoidance is accounted by the A* algorithm.

Introduction of multiple robots

- Multiple robots were introduced to the simulation.
- Each of the robots are managed by separate threads.
- This enables us to control the robot concurrently
- A counting semaphore is used to track the number of idle robots.
- The robot collision is handled by blocking all the path which are active.

Task allocation

- A task is defined as a tuple which should contain the pickup position, drop position and the package-id.
- The user can add task either through web-app or an API.
- The task is processed in first come first serve manner(Queue).

Algorithm - Task allocation

Algorithm 1: Task Allocation

```
pool = ThreadPool(number of robots);
semaphore = CountingSemaphore(number of robots);
q = Queue();
mutex = Mutex();
while True do
    Wait for an task to arrive in the queue.;
    Acquire the mutex.;
    Try to acquire the counting semaphore.;
    Spawn a thread from the threadpool.;
    Find a idle robot.;
    Release the mutex.;
    Do the task. ;
    After the task completion, release the counting semaphore.;
end
```

Algorithm - Task execution

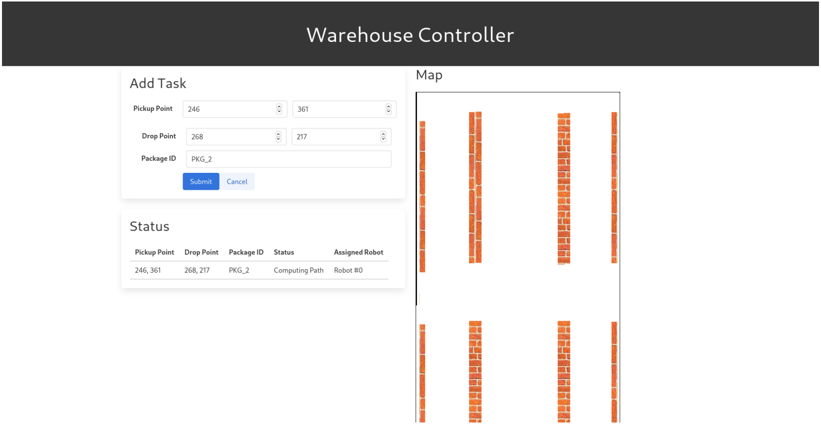
Algorithm 2: Task Execution

A task should contain the coordinates of pickup and drop position and the package id;
Path to pickup and drop position is calculated.;
if *Path not found* **then**
    quit;
end
Track the path to the pickup position;
Load the object to the roof box;
Track the path to the drop point;
Unload the object.;

User Interface

- A web app was developed to interact with the system.
- Users can access the web app through any device which have a web browser installed.
- User can add task by providing the pkg-id and cooridinates.
- The coordinates can be selected from the embedded map in the web app.
- After the submission of task, the user is updated with the status of task.
- Web socket technology is used to provide realtime updates to the user.
- The user can create their own user interface with the help of provided web API.

Web app


Warehouse Controller
Add Task
Pickup Point
246
361
Drop Point
268
217
Package ID
PKG_2
Submit
Cancel
Status
Pickup Point
Drop Point
Package ID
Status
Assigned Robot
246, 361
268, 217
PKG_2
Computing Path
Robot #0
Map

API Description - HTTP

/getTasks: This endpoint is used to get information of all tasks in the system. It returns a JSON object.

```
[
 { uuid: 243252153,
  pickup: {x: 134, y: 42},
  drop: {x: 23, y:324},
  package: ”pkg-1”,
  status: ”Finished”,
  pickupPath: [[3,5], [3, 6] ...],
  dropPath: [[3,3], [3, 1] ...],
 }
]
```

API Description - HTTP

/getRobotsPos: This endpoint is used to get current position of all robots in the system. It returns a JSON object.

```
[
 {
  id: 5,
  x: 65,
  y: 234,
  angle: 34
 }
]
```

API Description - WebSocket i

1. **addTask -** This event should be emitted to add a task to the queue.

```
{
  uuid: 243252153,
  pickup: {x: 134, y: 42},
  drop: {x: 23, y:324},
  package: ”pkg-1”,
}
```

API Description - WebSocket ii

2. **addRobot -** This event should be emitted to add a new robot to the system.

```
{
  id: 32
}
```

3. **updateStatus -** This event is emitted by the system when a robot updates its status. Different status available are:
    3.1 Queued
    3.2 Computing Path
    3.3 In Transit
    3.4 No Route
    3.5 Finished

```
{
  uuid: 334234,
  status: "In Transit",
}
```

API Description - WebSocket iv

4. **UpdatePath -** This event is emitted by the system after the path finding completes.

```
{
  uuid: 33451,
  pickupPath: [[3,5], [23,2]],
  dropPath: [[3,5], [23,2]],
}
```

API Description - WebSocket v

5. <u>AssignTo-</u> This event is emitted by the sytem when a task get assigned to a robot.

```
{
  uuid: 23425,
  robot: 34,
}
```

Demo

Questions?

References i

[1] P. R. Wurman, R. D'Andrea, and M. Mountz, "Coordinating Hundreds of Cooperative, Autonomous Vehicles in Warehouses",AIMag, vol. 29, no. 1, p. 9, Mar. 2008.
[2] T. Hellstrom and O. Ringdahl, 'Follow the Past: a Path Tracking Algorithm for Autonomous Vehicles', International Journal of Vehicle Autonomous Systems, vol. 4, no. 2/3/4, pp. 216–224, 2006.

[3] Marshall, Joshua and Barfoot, Timothy Larsson, Johan. (2008). Autonomous underground tramming for center-articulated vehicles. J. Field Robotics. 25. 400-421. 10.1002/rob.20242.

[4] N. A. M. Johari, H. Haron and A. S. M. Jaya, "Robotic modeling and simulation of palletizer robot using Workspace5," Computer Graphics, Imaging and Visualisation (CGIV 2007), Bangkok, 2007, pp. 217-222, doi: 10.1109/CGIV.2007.73.

[5] R. Caccavale et al., "A Flexible Robotic Depalletizing System for Supermarket Logistics," in IEEE Robotics and Automation Letters, vol. 5, no. 3, pp. 4471-4476, July 2020, doi: 10.1109/LRA.2020.3000427.

[6] Berenson D, Srinivasa S, Kuffner J. Task Space Regions: A framework for pose-constrained manipulation planning. The International Journal of Robotics Research. 2011;30(12):1435-1460. doi:10.1177/0278364910396389

[7] R. Varga and S. Nedevschi, "Vision-based autonomous load handling for automated guided vehicles," 2014 IEEE 10th International Conference on Intelligent Computer Communication and Processing (ICCP), Cluj Napoca, 2014, pp. 239-244, doi: 10.1109/ICCP.2014.6937003.

Thank You