Group 2
Final Project

Evan John Gonnerman Connor L Yates Jiongcheng Luo Yeang Kong Zheng

Exploration Problem:

Problem:

- Unknown map
- Unknown global position

Solution:

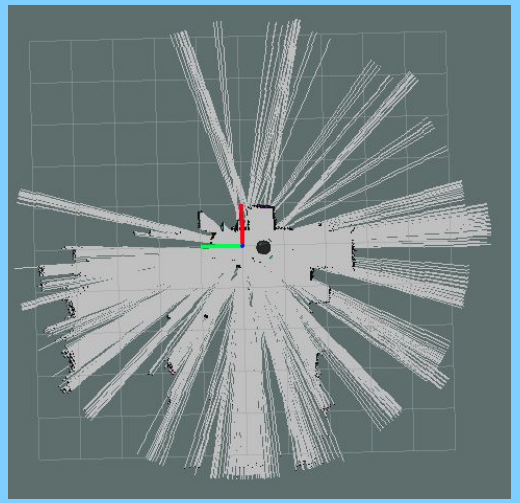
- SLAM (Simultaneous Localization and Mapping)
- Laser scan running gmapping returns occupancy grid
- Navigation running nav_bundle yields navigation commands

Packages:

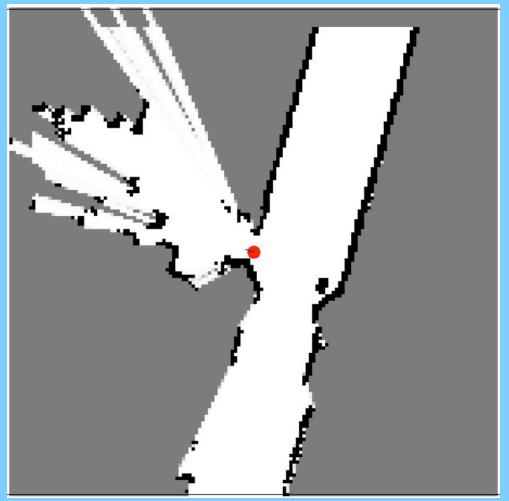
- Gmapping:
  - Output → Occupancy grid with values -1, 0-100
  - -1 Value = Cell has not been seen
  - 0 Value = Cell has been verified as free
  - 100 Value = Cell has been verified as occupied
- Nav_bundle:
  - Output → Odometry data, velocity control
  - Use odometry to track position
  - Use Twist commands to yield linear/angular velocities

-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1
-1	30	60	70	50	92	-1
-1	10	20	45	12	80	-1
100	0.5	0	0	0	15	100
100	0	0	0	0	0	100
100	0	0		0	0	100

Occupancy grid example

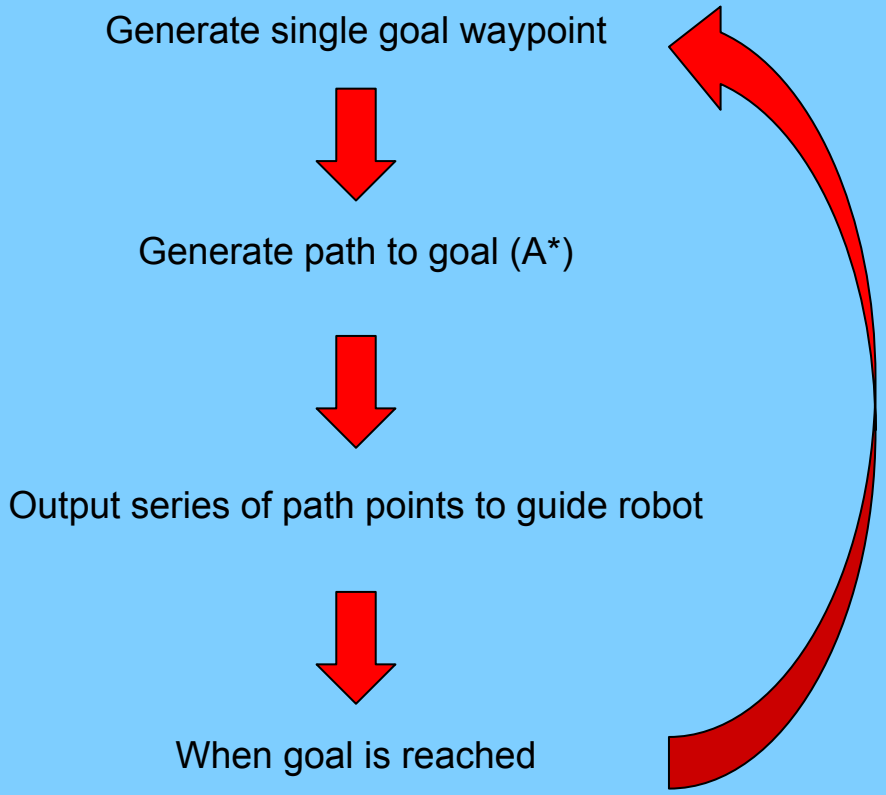


Laser scan example



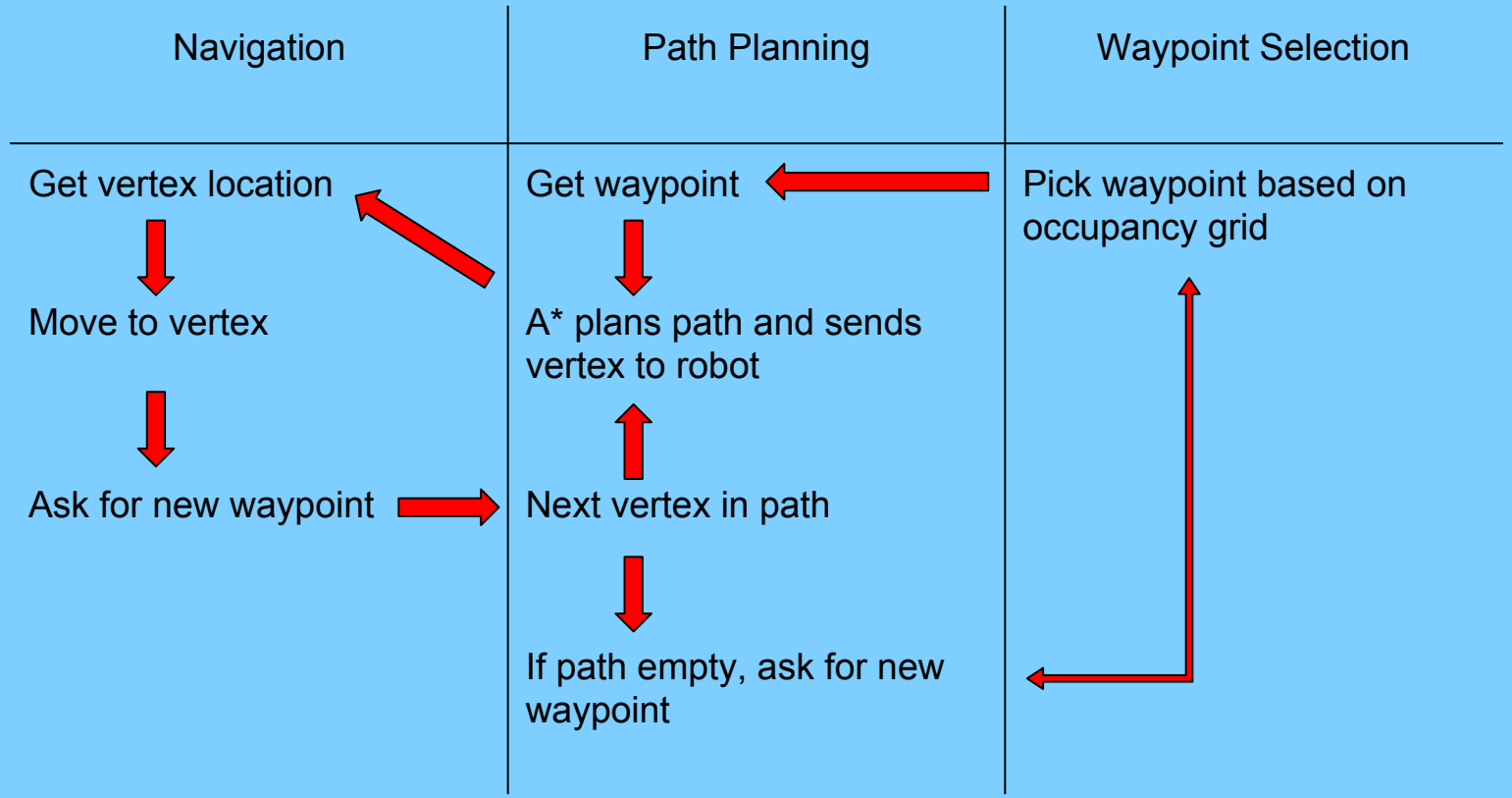
World image example

Steps:


Generate single goal waypoint
Generate path to goal (A*)
Output series of path points to guide robot
When goal is reached

Scripts:

- Waypoint Selection:
  - Waypoints will be selected as free cells adjacent to unknowns cells
    - Cell value of 0 adjacent to cell value of -1
  - Waypoint x-y sent to path planning script
- Path Planning:
  - Utilizes A* as path planning algorithm
  - Takes current position and waypoint
- Navigation:
  - Takes current location → Moves to vertex of path planning
    - Current location = Based off odometry data
    - Velocity command sent to base_link_goal


Navigation
Path Planning
Waypoint Selection
Get vertex location
Move to vertex
Ask for new waypoint
Get waypoint
A* plans path and sends vertex to robot
Next vertex in path
If path empty, ask for new waypoint
Pick waypoint based on occupancy grid

Success:

- Waypoint Selection:
  - Runs frontier based search for viable waypoint
  - Publishes and subscribes with A* path planning algorithm
- A* Path Planning Algorithm:
  - For given current location and goal location plans path
  - Outputs set of vertices
- Navigation:
  - Moves from current x, y location to vertex from A* path plan
  - Publishes and subscribes to A* path planning

Problems:

- Communication:
  - Talking between nodes presented multiple challenges
  - Time-stamped data types vs. boolean/float/point data types
  - Calling of time synchronized functions
- Bringing the whole package together:
  - Issues arose when different messages were passed
  - Hard coding vs. variable messages

Who Did What?

- Connor Yates
  - Frontier Waypoint Selection Algorithm
  - ROS Master
- Jiongcheng Luo
  - A* Path Planning Algorithm
- Evan Gonnerman
  - Navigation Algorithm
  - Slides/Presentation
- Kong Zheng Yeang
  - Slides/Presentation

Summary:

- Abilities:
  - Could hard code waypoints for robot to travel to in worlds
  - Could hard code A* to plan a path based off of known position and goal
  - Could pick viable waypoint with frontier based search algorithm
  - Could talk between all scripts/topics
- With More Time:
  - Rewrite callback function for path planning
  - Address issues in communication between nodes

Questions?