Particles: struct with fields x, y, theta, w -> N-by-3 matrix [x, y, theta]. Weights themselves are an array.

Begin function

* [map, particles] = init(map\_file, num\_particles)
* [sensor\_data, laser\_data, odometry\_data] = getSensorData(data\_file)

loop

* [prediction, observation] = predict(particles, two\_consecutive\_laser\_readings)  
  or [prediction, observation, end\_idx] = predict(particles, sensor\_data, start\_idx)
* \*\*\* compute weights (w)

for each particle p\_i

w\_i = 1

for each laser beam angle

get grids on line (until touch -1)

get prob of walls

compute distance from possible walls

w\_i \*= sum(p(wall) \* look up sensor model (undetermined))

end for

end for

* new\_particles = resample(particles, weights, num\_samples)
* visualize(map, robot\_pos, particles)

end loop

End function

Documentation

**[map, particles] = init(map\_file, num\_particles)**

Initialize the map, generate initial samples, and visualize. Samples are drawn with the function randsample(population, num\_samples, true, prob).

Output:

* map is a struct with the following fields:

1. resolution, offset\_x, offset\_y (for beesoft maps; not necessary for this project)
2. size\_x, size\_y: full size of the map
3. min\_x, max\_x, min\_y, max\_y: the min/max x/y coordinate values of known grids (grids whose prob ~= -1)
4. prob: a matrix with size size\_x by size\_y

* particles is num\_particles-by-3 matrix; particles(i,:) = (x, y, theta) coordinates of a particle

**[sensor\_data, laser\_data, odometry\_data] = getSensorData(data\_file)**

Input:

* data\_file: file name of sensor data log file

Output:

* sensor\_data: a struct array with the following fields:

1. timestamp
2. type: ‘L’ (laser) or ‘O’ (odometry)
3. robot\_pos: (x, y, theta)
4. laser\_pos: (xl, yl, thetal) if type == ‘L’
5. laser\_reading: array of readings if type == ‘L’

* laser\_data: matrix of laser readings (sorted by timestamp)
* odometry\_data

**[prediction, observation] = predict(particles, laser\_readings)**

Reference: http://www.cim.mcgill.ca/~yiannis/particletutorial.pdf

Input:

* laser\_readings: laser\_readings at time t and t+1

Output:

* observation: laser data (excluding robot pos and timestamp) at time t+1

**[prediction, observation, end\_idx] = predict\_with\_all\_sensor\_data(particles, sensor\_data, start\_idx)**

Input:

* sensor\_data: struct
* start\_idx: the index of sensor\_data at time t

Output:

* end\_idx: the index of the first laser data

**new\_particles = resample(particles, weights, num\_samples)**

Input:

* weights: unnormalized weights
* num\_samples: optional; default value is size(particles, 1)