SugarTrail Code Introduction

1 Phase 1: Signature Test

- Location:
  - Getting Data: \sugar\cluster\reading_testing
  - Analysis: \sugar\cluster\analysis\lab_signature
- Fig:
  - Stable reading for one node
  - Disturbed reading for one node
  - Relatively stable signature for one node
- Run: \sugar\cluster\analysis\lab_signature\sig_analisis.m

2 Phase 2: Distribution Table/Curve generate

- Location: \sugar\cluster\distribution
- Fig:
  - Relation of ranging distance and real distance
- Run:

3 Phase 3: Hallway Test

- Location: \sugar\cluster\hallway_simulation
- Fig:
  - Cluster
  - Navigation
  - Base number and performance
  - Cluster size threshold and performance
- Run: main.c

4 Phase 4 : Lab(Field Test)

- Location: \sugar\cluster\lab_field_test
- Fig: None
- Run: main.c

5 Phase 5: Lab(Database Test)

- Location: \sugar\cluster\lab_database
- Fig:
  - Clustering(path_length = 10000)
  - Navigaion
- Run:
  - p1_collect_readings.m (collecting data, done, don't run again)
  - p2_data_process.m (generate proper data structure for later, done, don't run again)
  - p3_generate_cluster.m
  - p4_match_area_to_cluter.m
  - p5_testing.m
  - p6_kmeans.m

6 Phase 6: Supermarket Experiment

- Location: \sugar\cluster\supermarket_exp
- Supermarket layout: supermarket.skp
- Data:
  - Raw Data: \sugar\cluster\supermarket_exp\step3\raw_data\
  - Complete Data and Documentation: \sugar\cluster\supermarket_exp\step3\complete_data\
- Fig:
  - Compass Direction
  - Clustering(path_length = 10000)
  - Navigation
- Run:
  - p1_collect_readings.m (collecting data, done, don't run again)
  - p2_data_process.m (generate proper data structure for later, done, don't run again)
  - draw_compass_direction.m
  - p3_generate_cluster.m
  - p4_testing.m
- Code for analysis: \sugar\cluster\supermarket_exp\analysis
  - .m files:
    - angle_convert.m : convert direction information into angle
    - dijkstra.m : Dijkstra Algorithm
    - direction_convert.m : convert current reading into direction info based on the reading at that point
    - draw_cluster.m : draw the clusters, should be updated with translucent style soon
    - draw_compass_direction.m : draw the magnetic field based on compass reading in supermarket

    * get_cluster_sig.m : get the cluster which the signature belongs to, used to implemented by choosing the minimum eclidean distance, but with the using of sub-set in signature, we change it to use possibility also, but it does not work well with the later one, cannot select correct cluster often.
    * get_next_point.m : used in the "training part", when the training paths are generated, it will give out the next point in path, based on random walk and the data from supermarket.
    * get_next_step.m : used in the "testing part", when the guiding paths are generated, it will give out the next step location in navigation, based on guiding info and the data from supermarket.
    * guide.m : the main part of guiding
    * is_blocked.m : check whether if the path of current point and next point is blocked by environment.
    * navigate.m : called in testing.m for navigation from one point to another
    * possibility.m : give out the possibility of one ready belongs to certain cluster.
    * valid_sig.m : check whether the signature reading is valid and how many valid info is in one signature.
- .mat files:
    * 2feet_grid.mat : real supermarket readings, ranging with different anchors and compass reading at each point of the grid
    * processed_data.mat : the data processed from raw data.
    * distribution_table.mat : calcutate the possibility of one reading belongs to the cluster with certain center
    * environment_info.mat : racks' edges info, for obstacle detection