-Mapping	⇔	-Localization
------------------		---------------------------
Knowing the surroundings of the robot		identify the robot, given all the sensor data
"mind mapping"		-depends on the data being represented
- but requires knowledge about its current location		|→a distance measure of its surroundings**
- **recognizing** the word data (not dead reckoning)		|→ identify objects using visual cues
---		- requires a map to perform any of the above

Summary: know the position of the object and the relative location of the same (distance)

-- --

|-> basic objectives:

-> perceive the various entities in the world
-> define a marker for each of the same (characteristic feature)
-> very basic: obstacle (not accessible)
free space (accessible)
not known (occluded and unknown)

methods

- Online map (have to be controlled initially)
  - Navigate the area manually (generally tele-operated) and construct the map -> used in emergency situation (e.g., drones during evacuation efforts)
    - In the first instance data is understood only by the human operator
    - For autonomous navigation within the same environment
      - identify the objects in the map, defining a cost (weight of the entities) map and train the system identify the objects (field of view is limited, everything is new, for all t= 0,1,2…..n)
  - doesn't support autonomy since deployment and autonomy in unknown environment
  - localization is postponed to set of tasks post the map generation
  - Moreover, limited training sets restrict its perception ability
  - Examples: gmapping, rgb-d slam etc.

    ------------extent of human intervention needed: **HIGH**
- Active recognition (**novelty** in the current work)
  - Features: minimal sensor coverage (aesthetics-design constraint) LRF ~ 120^o & prime sense ~ 40^o, but still constructs the 360^o perception map (online).
  - Classifies segments within the map to be either obstacles or free space or not known
  - Defines a cost map for the complete spatial data, localization inherent to map construction
  - Standalone system and supports data(depth) transmission over minimal band-width
  - Compatible with any system which can perceive 3D spatial data & publishes odom data
  - Supports information extraction from any point within the

Design Standards

- Consumer grade hardware (Asus Xtion Pro Live RGBD, Our PWC is a Quickie R Rhythm™ with custom rotary encoders , i7-3720QM processor, 8GB RAM and a 120GB SSD)

- ➔ Real Constraints (based CanWheel, Wizard of Oz study)
- ➔ Standard mathematical modelling techniques (C++, PCL library, Probabilistic techniques)
- ➔ Accepted Data transfer protocol (CANBUS information protocol, registered data structures)
- ➔ Hardware and Software (ROS_ hydro version on Ubuntu 12.04 LTS)
- ➔ Algorithm standards (computational efficiency, ease of implementation, reproducibility)

Design Process

- ➔ Mapping algorithms:
    - o Worked on:
        - ▪ Gmapping: requires a map server, and so **could be used** confined places
            - • map can updated during the robot navigation
        - ▪ Rgb-d slam: computationally very expensive, *gmapping is preferred over rgbd-slam*
    - > Every object in the map is perceived to be same as every other objects in the map, the description of every objects has to be manually defined (**COST MAP**)
    - > Both the techniques work well with a pre-defined map (**map server**), relyin on a static map constrains the potential benefit of any new work based on such grounds
    - > Localization: dependent on the accuracy of the map server
- ➔ What is need for localization?
    - o To make the robot aware of its *presence within its surroundings*
        - ▪ *Entity$_n$ has this identity and its position is at (x, y, z)$_n$* relatively → **mind mapping**
    - o Local presence is the driving criteria
    - o Global presence to derive the navigation plan (guide the t+1$_{th}$ movement update)
    - o Local map + guide -> global presence (to carry out tasks e.g., move from Pt A to Pt B)
- ➔ Gold Standard for mapping: **OCCUPANCY GRID MAPPING**
- ➔ Because sensory perception is based on polar coordinate system, a POLAR map is easy to construct
    - o Circular maps (center point defines the location of the robot, entities within: surroundings)
    - o |-> Egocentric maps
- ➔ In an indoor environment, what are basic attributes that distinguish open and occupied space?
    - o ***Generally***: *<u>open space is navigable but occupied spaces refer to obstacles for navigation</u>*
    - o Navigable space: defines the presence of path, but mathematically?
        - ▪ Path: indicates a road to traverse -> (floor) Planar surface

Abstract

Mobile robots (smart powered wheelchairs) for indoor navigation offer the possibility of enhanced mobility to a large and growing population (most notably older adults), and a key feature of such a chair is obstacle recognition and avoidance. Sensors are required to detect nearby obstacles; however, complete sensor coverage of the immediate neighborhood is challenging for reasons including financial, computational, aesthetic, user identity and sensor reliability. However, direct modeling and control of commercial wheelchairs is not possible because of proprietary internals and interfaces (standard CANBUS protocols). In this work we design a dynamic egocentric occupancy map which maintains information about local obstacles even when they are outside the field of view of the sensor system on the basis of a probabilistic mapping scheme, thus serving as a task specific module. The end result of the SLAM algorithm is an egocentric polar occupancy grid map published as a point cloud.

Keywords: allotment of weights to 2D matrix which represents spatial data, recursive state estimation, obstacle detection, map estimation without full sensor coverage, bayes filter algorithm, markov assumption,