Indoor Autonomous Robotic Navigation System

Non-formal Definition

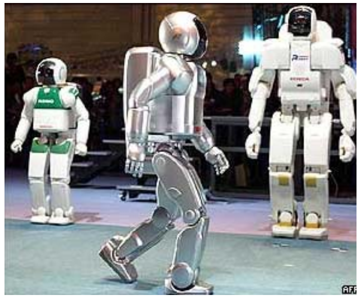
Robot can be defined as an embodied agent capable of perceiving the environment, form representations of it, make decisions and achieve certain objectives or goals. In recent years they have been asked to cooperate with other embodied agents!

Robots are of several kinds! Here are a few

Indoor Robot
The Nomad

Outdoor robot:
The Navlab

Dog Robot:
Sony AIBO



Humanoid
Robots: SONY

Interdisciplinary Features

- Mechanical Design, Construction
  - Ackerman wheel, differential drive, synchronous drive, legged robots, …
- Electronic design, hardware instrumentation
  - Sensors (sonar, laser rangefinder, IR), vision, communication (wireless, tether), …
- Algorithmic aspects
  - Algorithms for intelligent functionality / simulation of intelligent decision making

What Constitutes Autonomy

- Ability to move/navigate
  - To know where you are
  - To know what is around
  - To know where to go
  - To know how to go
  - To not bump around

What constitutes Autonomy

- To eat or drink when you want
  - Docking to charge, self charging capabilities

- To possess various locomotion modes. I walk, cycle, drive, fly
  - Reconfiguring abilities
  - Adaptive gaits
    - Terrain Sensing Abilities

What Constitutes Autonomy

- To be autonomous I need to predict, anticipate
  - Predict future states of other agents
  - Predict needs of other agents

- To cooperate, collaborate
  - Help and get helped

What Constitutes Autonomy

- Memorize, Learn and Innovate
- Think and conceptualize
- To entertain and get entertained
- …..
- ….
- And the list goes on

Autonomous Mobile Robot Navigation System

Localization / State estimation

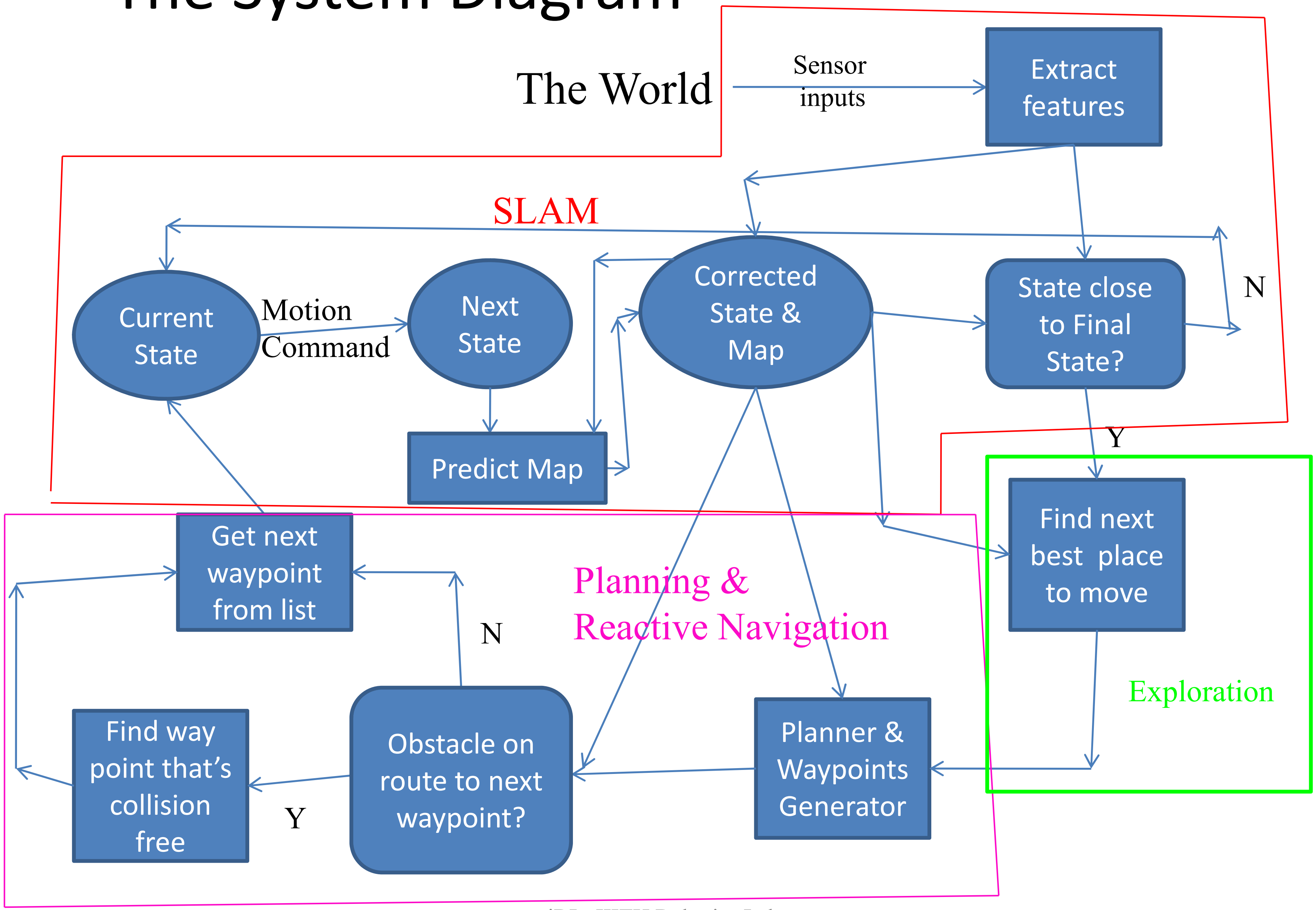
Reactive Navigation

Map building

Planning

Exploration

The System Diagram



The SLAM Problem

- The simultaneous localization and mapping (SLAM) problem asks if it is possible for a mobile robot to be placed at an unknown location in an unknown environment and for the robot to incrementally build a consistent map of this environment while simultaneously determining its location within this map.

- A solution to the SLAM problem has been seen as a “holy grail” for the mobile robotics community as it would provide the means to make a robot truly autonomous.

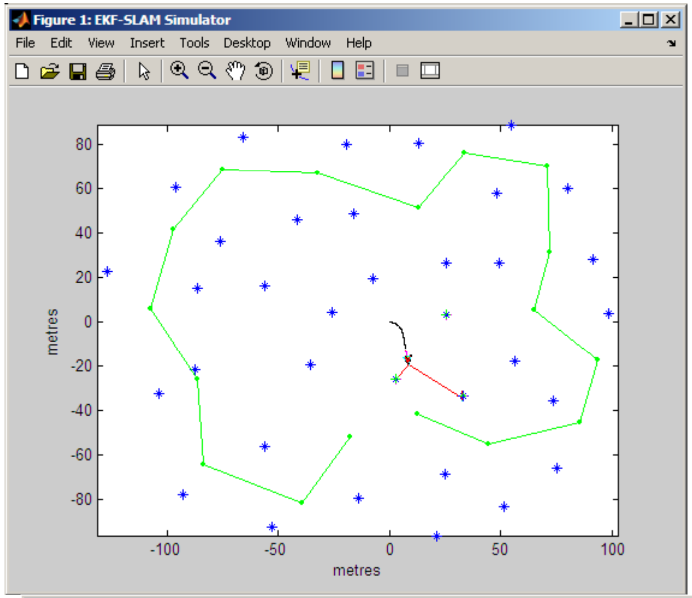
The SLAM Subsystem: Various Flavors

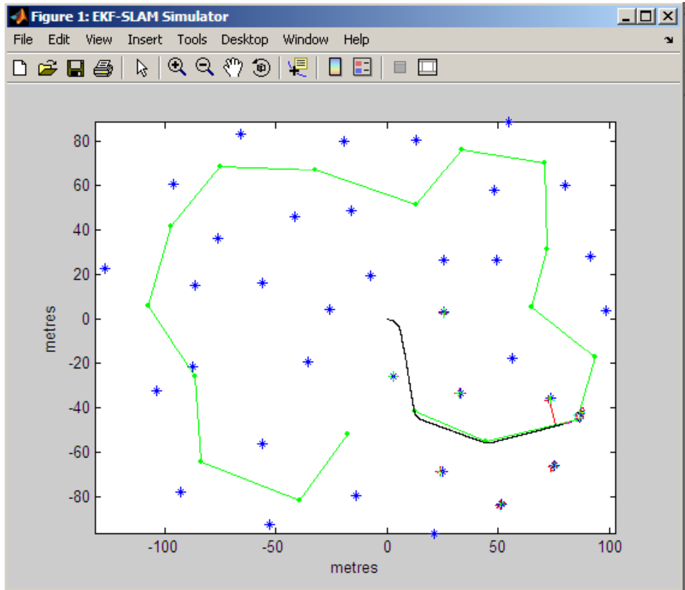
Popular Name	Inputs	Outputs	Sensing Modality
EKF SLAM	Set of map features (corners, line segments) seen so far, the last corrected robot pose , current motion command	A probabilistic filtered estimate of the current robot pose and all map features seen so far	Any range sensor (typically laser range finders)
Fast SLAM	Set of map features (corners, line segments) seen so far, history of motion commands	A probabilistic filtered estimate of the robot trajectory and all map features seen thus far	Any range sensor (typically laser range finders)
Graph SLAM	Set of map features (corners, line segments) seen so far, history of motion commands	A suboptimal estimate of the robot trajectory	Any range sensor

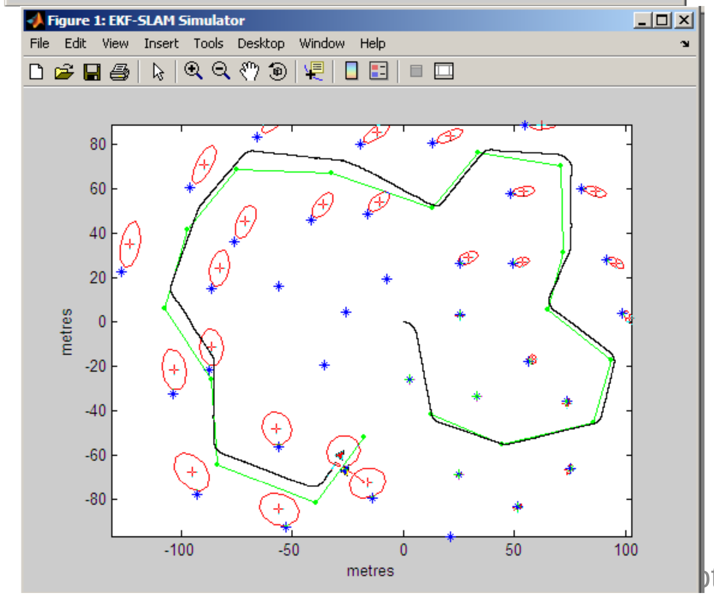
The SLAM Subsystem: Various Flavors

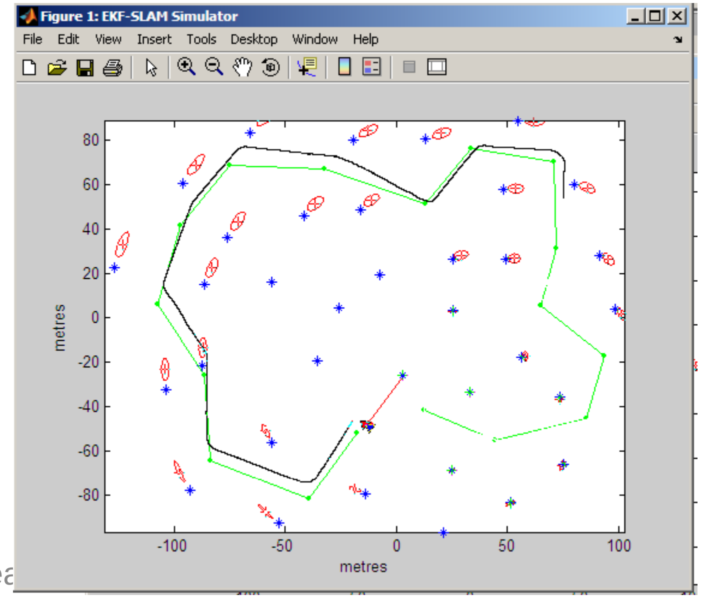
Popular Name	Inputs	Outputs	Sensing Modality
Mono-SLAM	A set of images/video and possibly an estimate of the most recent camera motion	Sparse 3D reconstructed points of certain image features and current camera motion up to a scale if there is no estimate of camera motion, up to a projective transformation if only a video from an unknown camera is given	Known or unknown monocular camera
Key-frame SLAM/ Structure from Motion	Set of map features (corners, line segments) seen so far, history of motion commands	3D reconstructed points of image features and camera trajectory found from an optimization framework generally up to a scale (known camera) or up to a projective transform (unknown camera)	Known or unknown monocular camera

EKF SLAM: Simulation Results

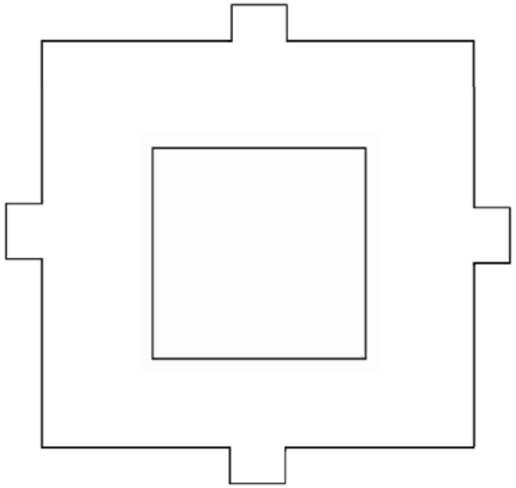


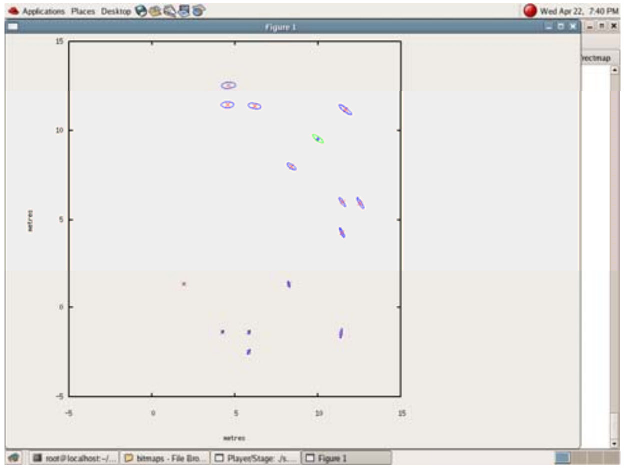


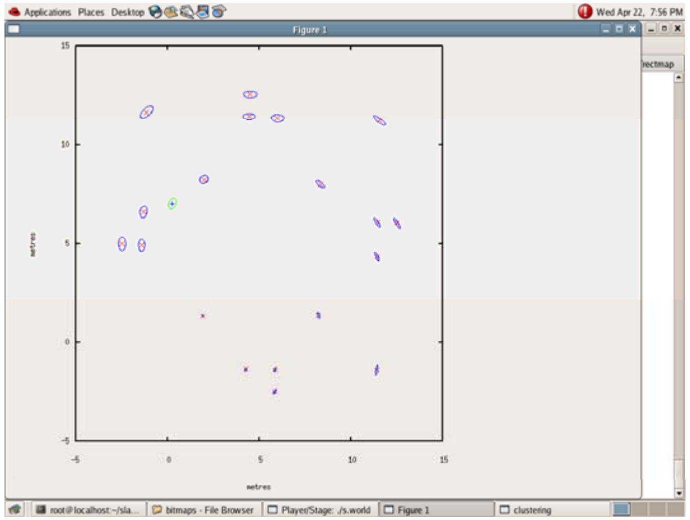


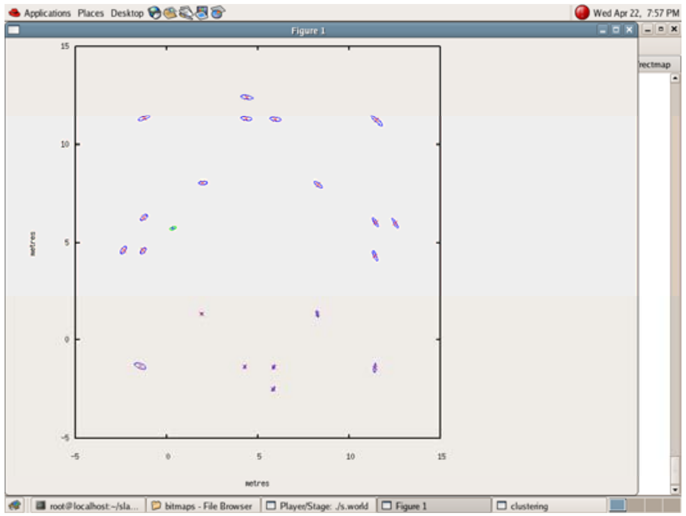


EKF SLAM : Realistic Maps



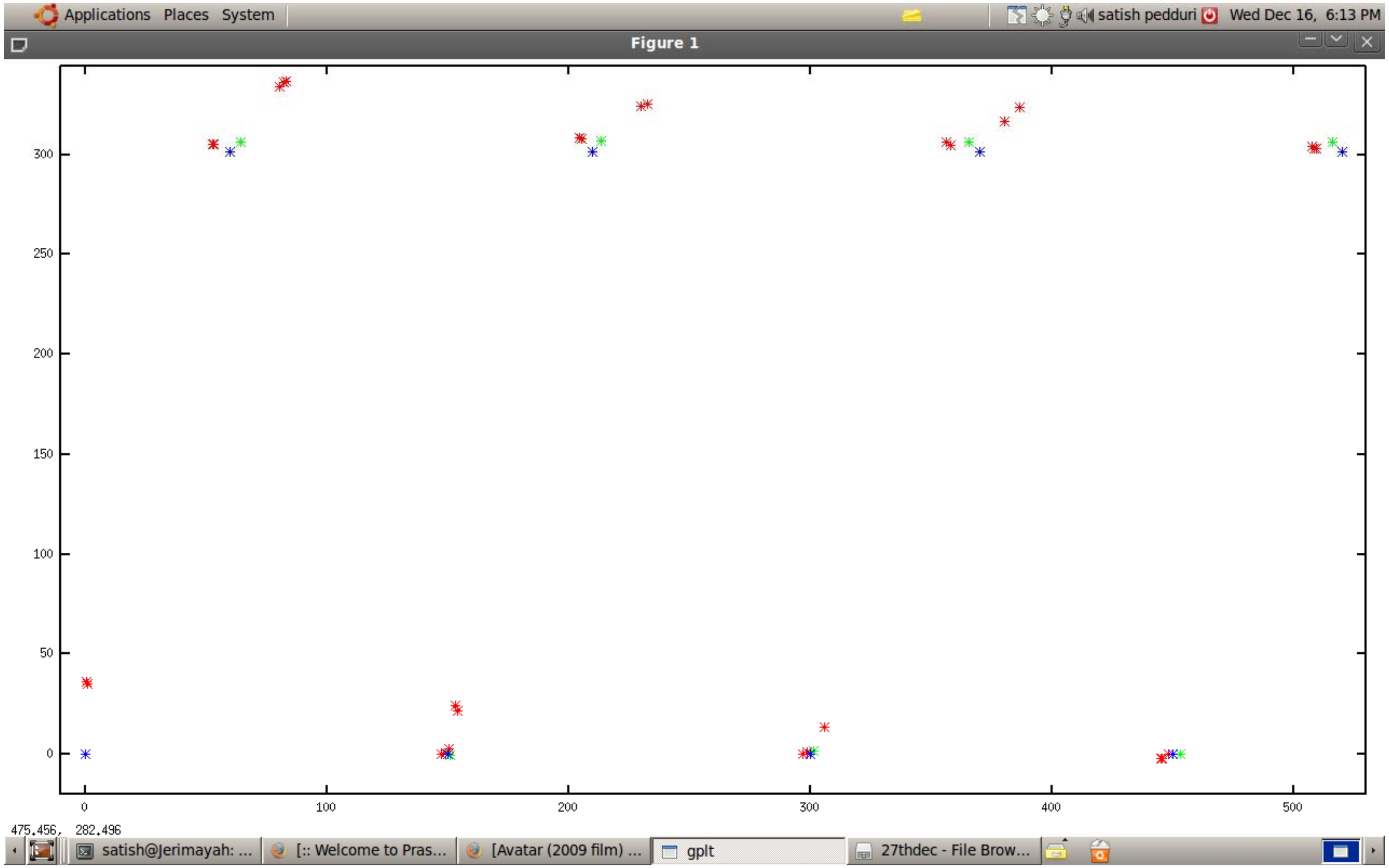






EKF SLAM Works!

May 09-: EKF SLAM Works!



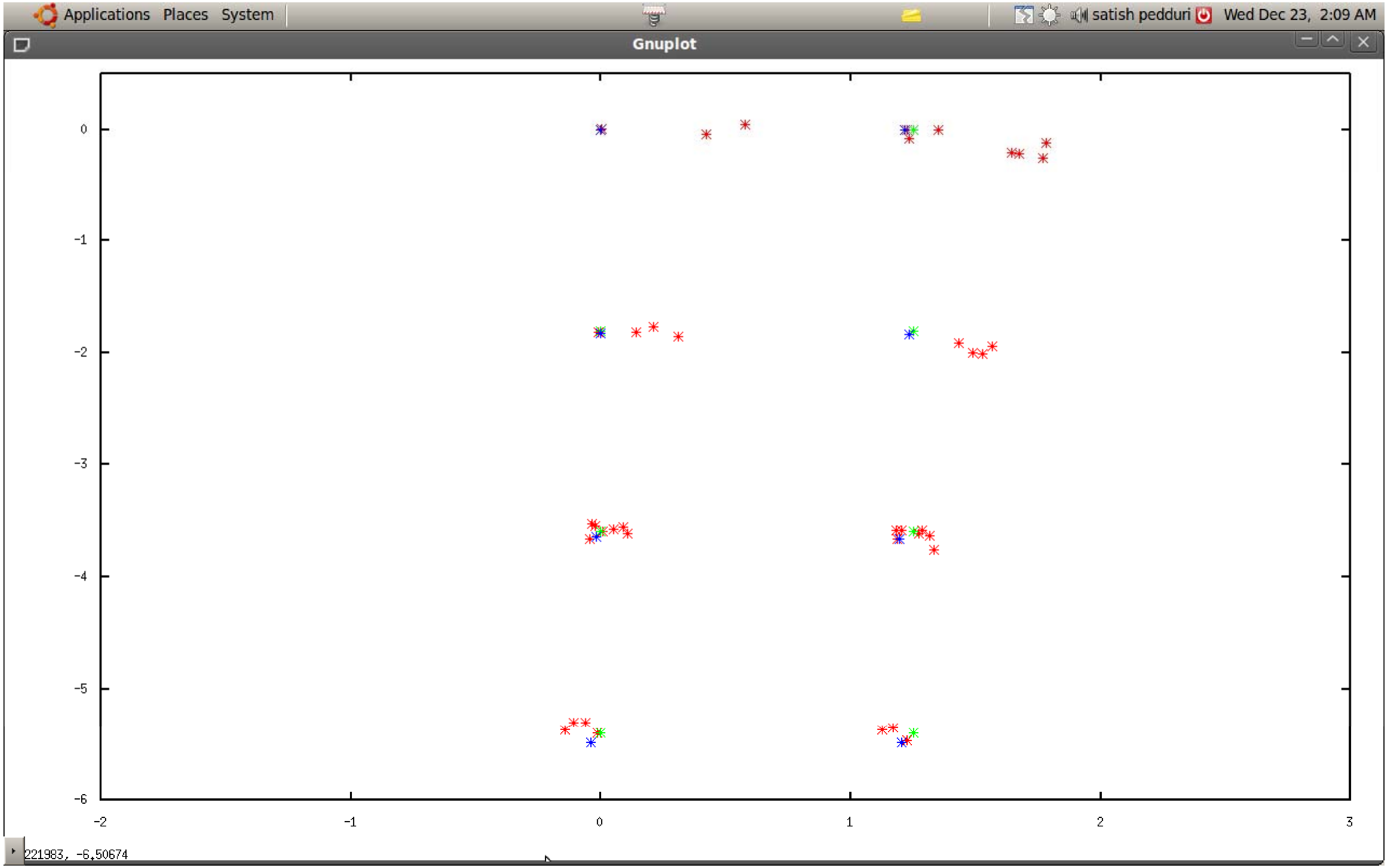
Red = Without SLAM

Green = With SLAM

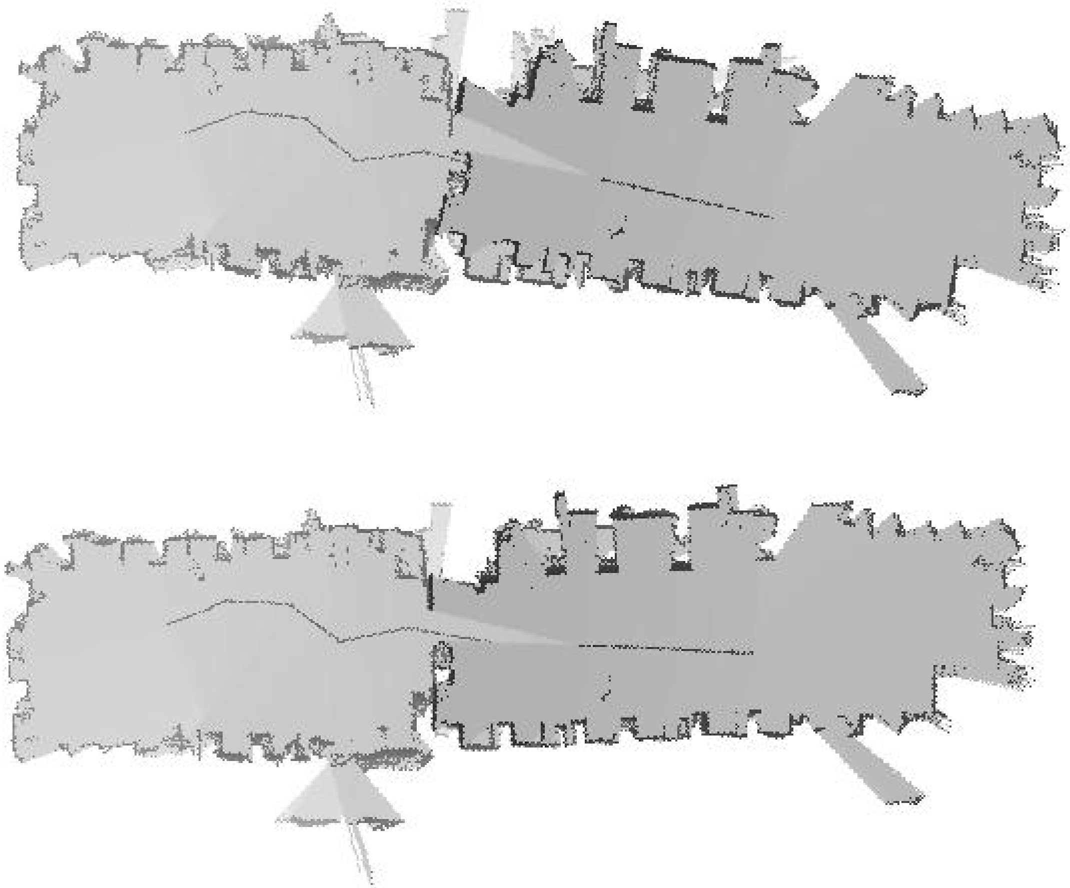
Blue = Ground truth box positions

May 09-: EKF SLAM Works!

May 09-: EKF SLAM Works!



SLAM vs No SLAM

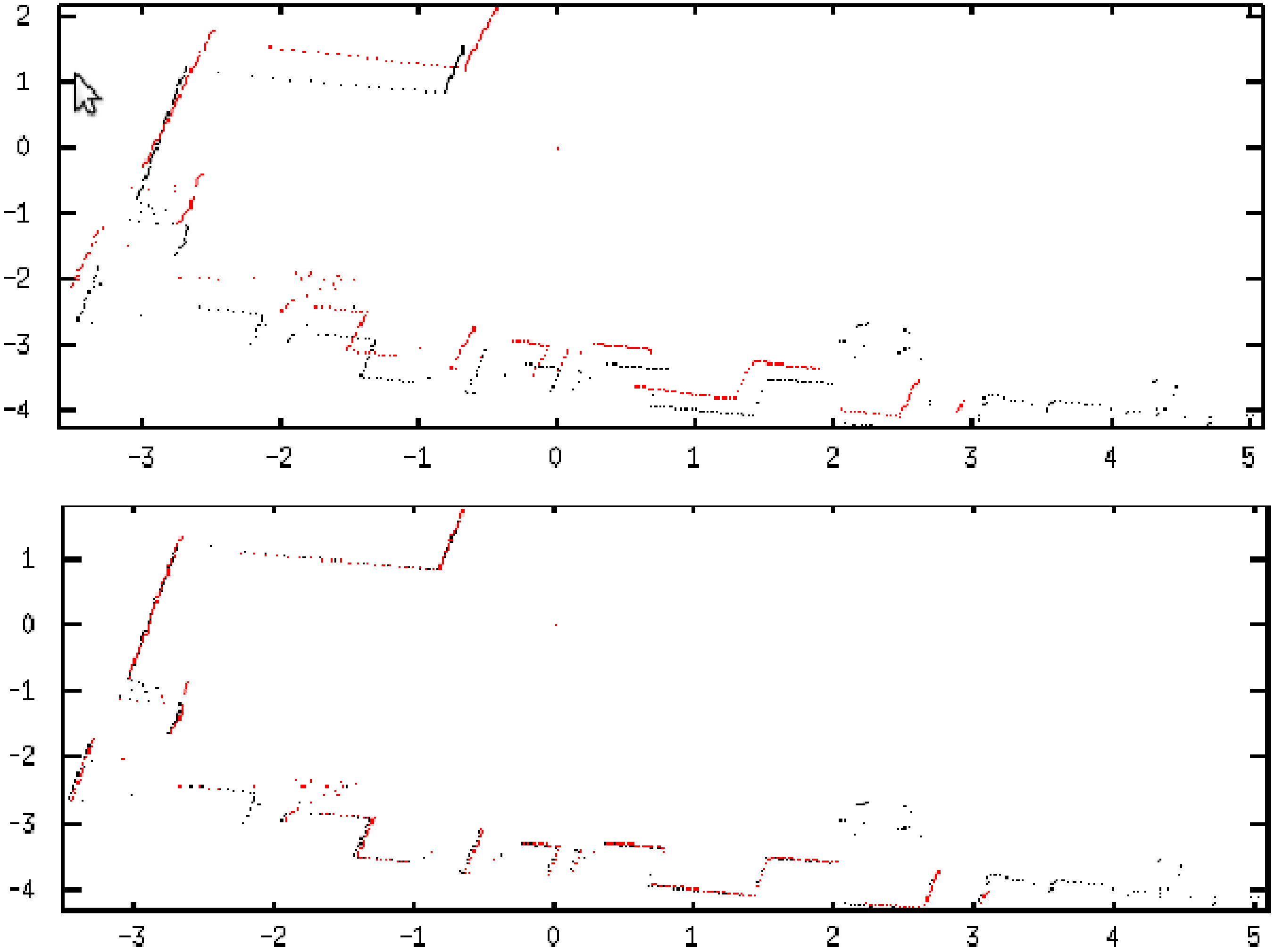


The top row shows results of a mapping process in a realistic lab environment without SLAM

The bottom row shows the map with SLAM.

The walls do not drift

Localized Forever: A 31 minute run



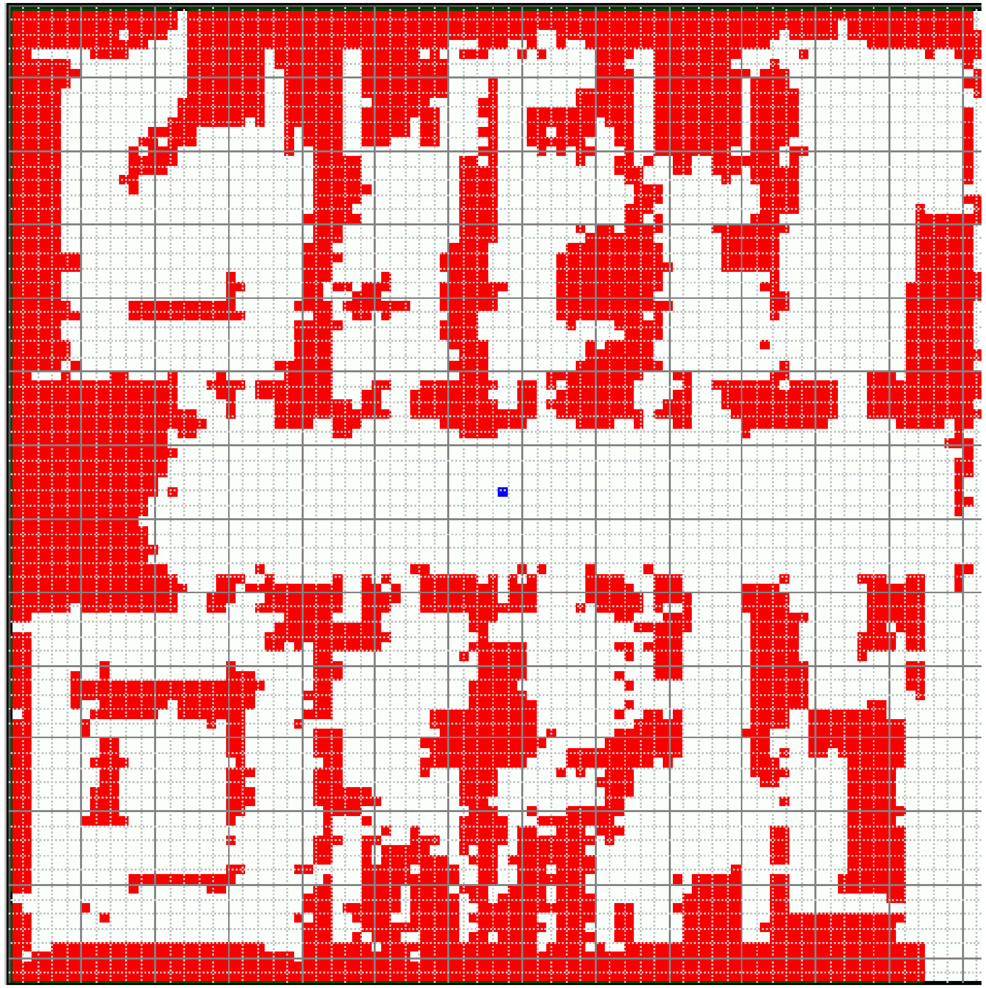
The top run is without SLAM

The bottom is a run in the lab for more than 30 minutes where the robot was always

State Estimation Subsystem

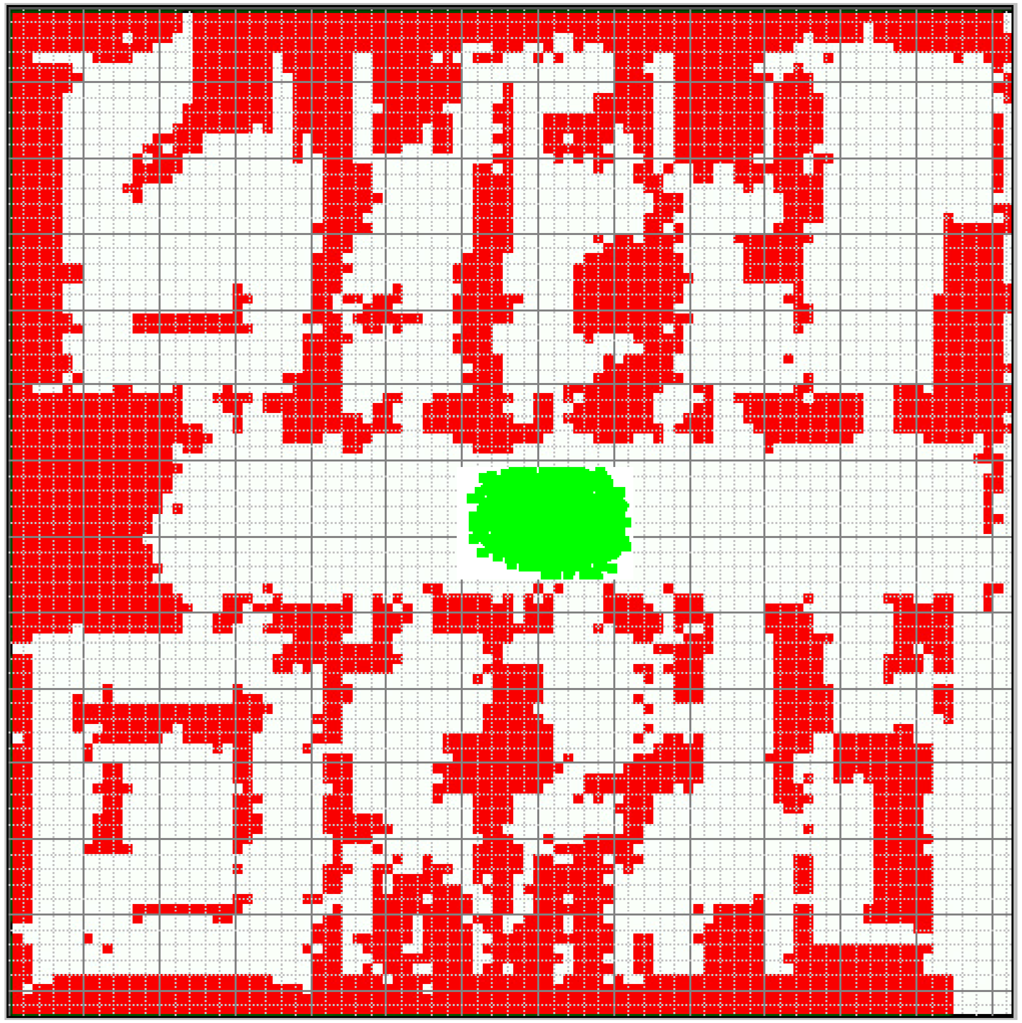
- When navigating in a known map state estimation is resorted to, i.e. estimating robot location or pose (location + orientation)
- Global State Estimation: When there's no initial guess of the robot's pose. Typically probabilistic techniques are used to come up with the most likely probabilistic distribution of the robot pose
- Local State Estimation: When there is an initial guess of the robot pose, which is erroneous and a less erroneous pose is computed through probabilistic as well as non probabilistic manner

Global State Estimation



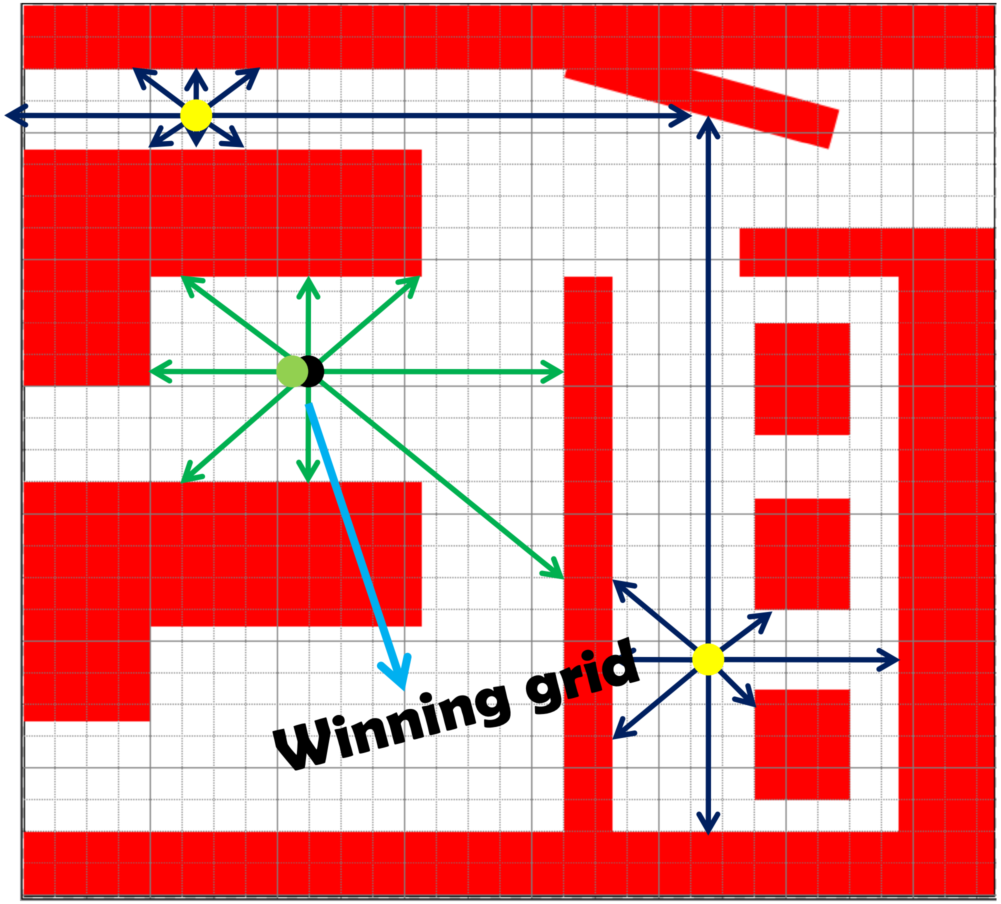
- Given:
  - Grid map
  - Robot Sensor readings
- To Find:
  - Robot's Pose
- Conditions:
  - No Initial estimate
  - Map is uncertain
  - Sensor Readings are uncertain

Local State Estimation



- Given:
  - Grid map
  - Robot Sensor readings
- To Find:
  - Robot’s Pose
- Conditions:
  - Initial estimate exists
  - Map is uncertain
  - Sensor Readings are uncertain

Markov Localization



$$Z = \{o_1, o_2, \ldots, o_m\}$$

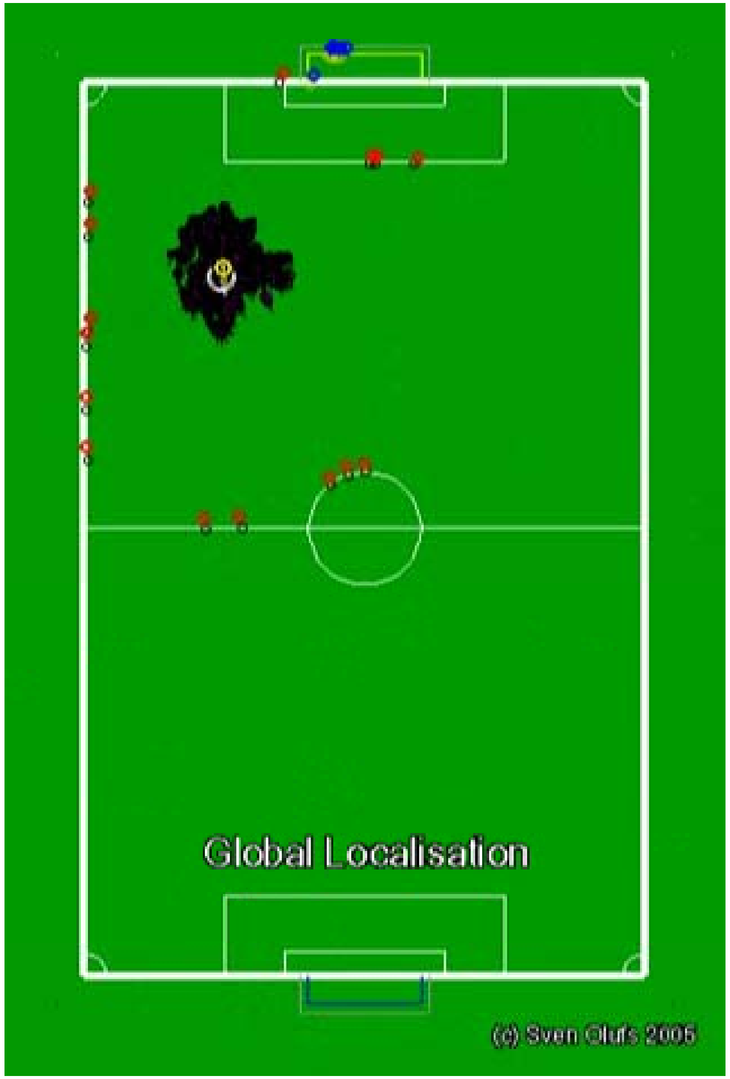
Algorithm

- For each free cell X in the map:

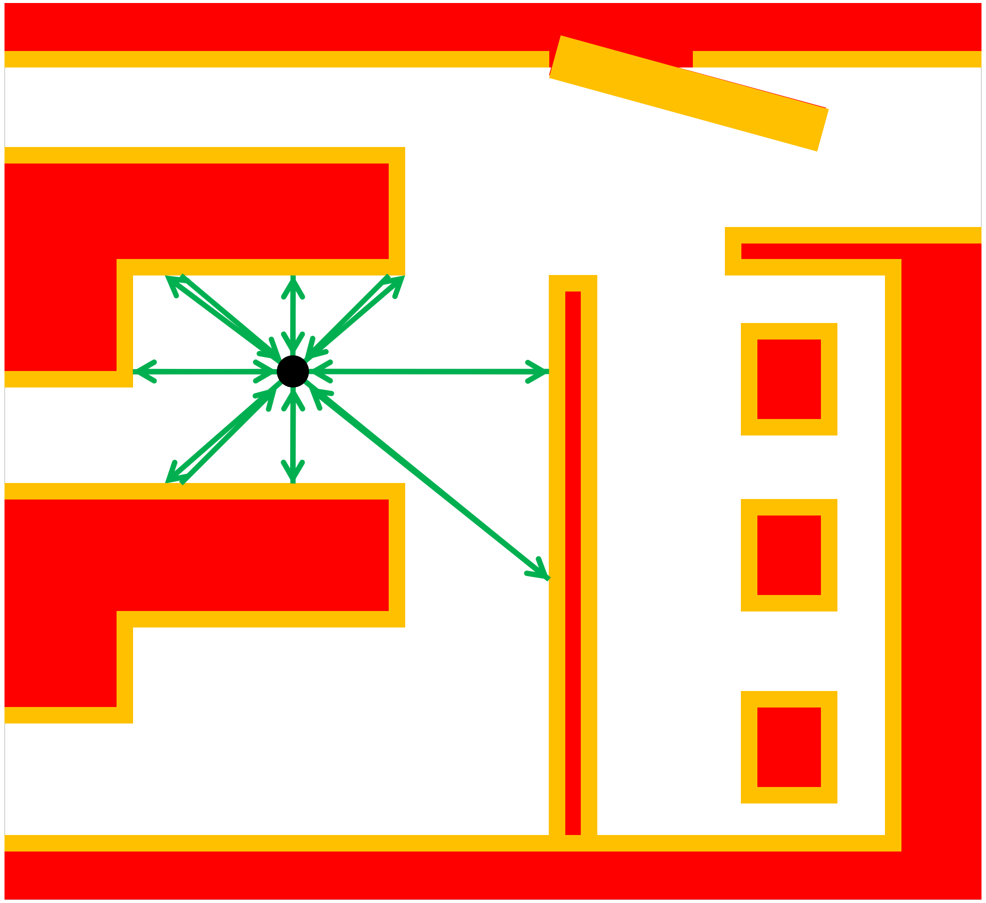
$$P(Z/X) = \prod_{i=1}^{m} e^{\left(\frac{(o_i - a_i)^2}{2\sigma^2}\right)}$$

- Normalize the probabilities
  - No. of computations $= \Omega(mn)$.
  - Precomputation of sensor readings

Global State Estimation: Monte Carlo



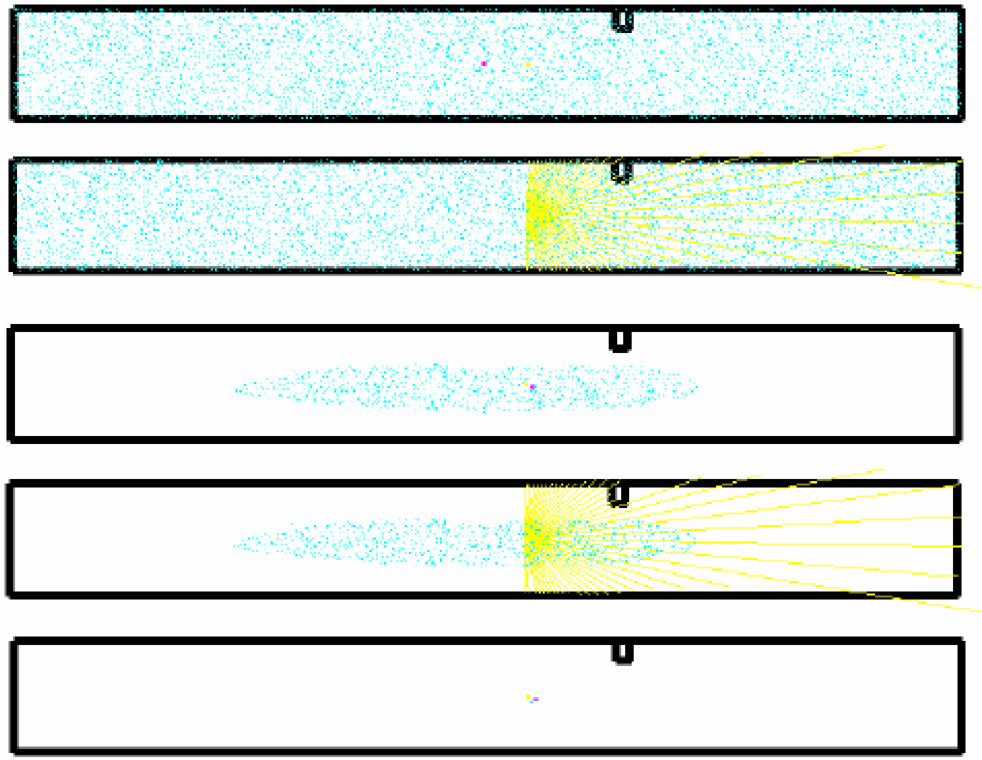
Global State Estimation: Reverse Projection



- The same sensor readings are re-projected from the boundary cells onto the free space.
- Place where all the *m* sensor re-projections meet is the state.
- Computations reduced to only boundary cells instead of entire free space.

Hemanth Korrapati and K Madhava Krishna, "Global Localization of Mobile Robots by Reverse Projection of Sensor Readings",
IEEE International Conference on Robotics and Bio-mimetics, 2008,
IEEE International Symposium on Measurement and Control in Robotics, 2008.

State Estimation Subsystem

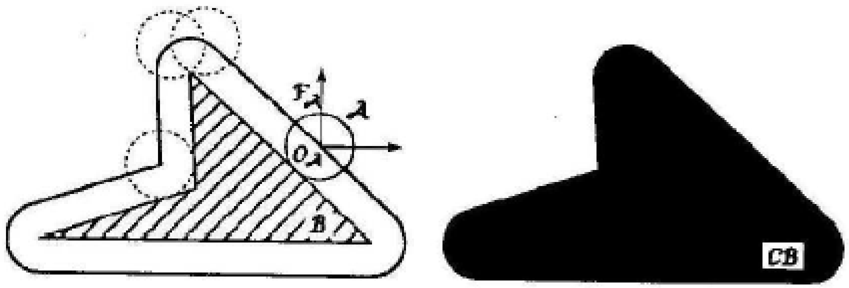


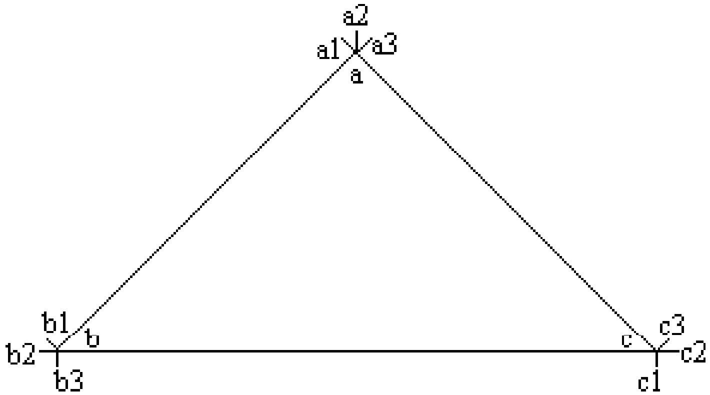
Planning Subsystem

Name of the Planner	Inputs	Outputs
Geometric Planner	A map such as a bit map or a map of line segments, start and goal poses of the robot	A collision free path that connects the start and goal pose of the robot. Under some very restricted conditions the path could be optimal
Kinematic Planner	Same as above	A collision free and kinematically consistent path
Kinodynamic Planner	Generally a 3D map, with terrain properties such as texture, coefficient of friction etc generally denoted as terrain mechanics	A collision free, kinematically consistent path and a set of velocities that provides for dynamic stability of the vehicle such that it obeys friction cone constraints, rolling without slipping constraints

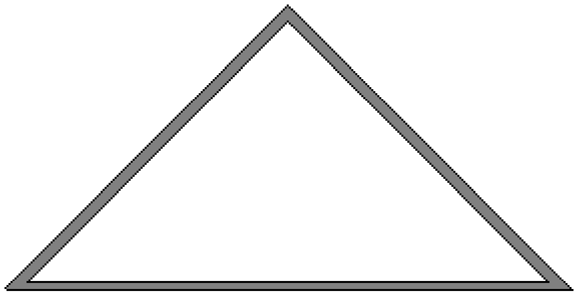
Planning: Computing the Configuration Space

- For the simple case of a circular robot



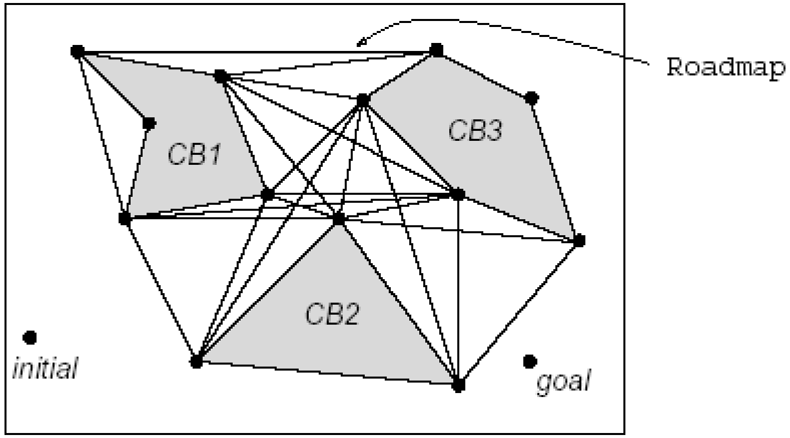


Points for the convex hull



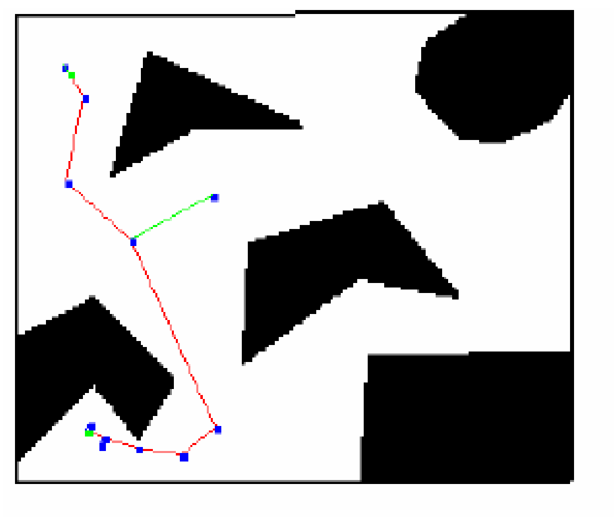
Simplification that you would use for
your algorithms

Planning: Constructing the Visibility Graph

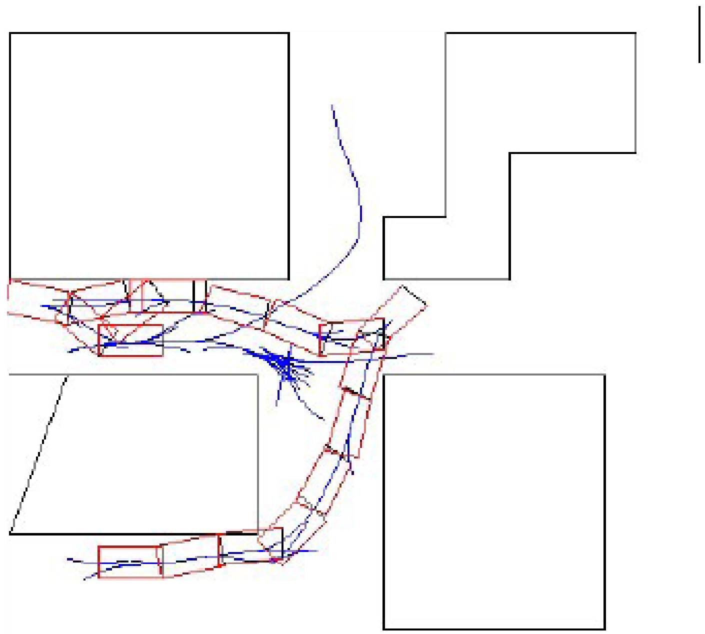


The visibility graph is an undirected graph G = (V,E) where the V is the set of vertices of the grown obstacles plus the start and goal points, and E is a set of edges consisting of all polygonal obstacle boundary edges, or an edge between any 2 vertices in V that lies entirely in free space except for its endpoints. Intuitively, if you place yourself at a vertex, you create an edge to any other vertex you can see (i.e. is visible).

Kinematic Planner



For a differential steer drive

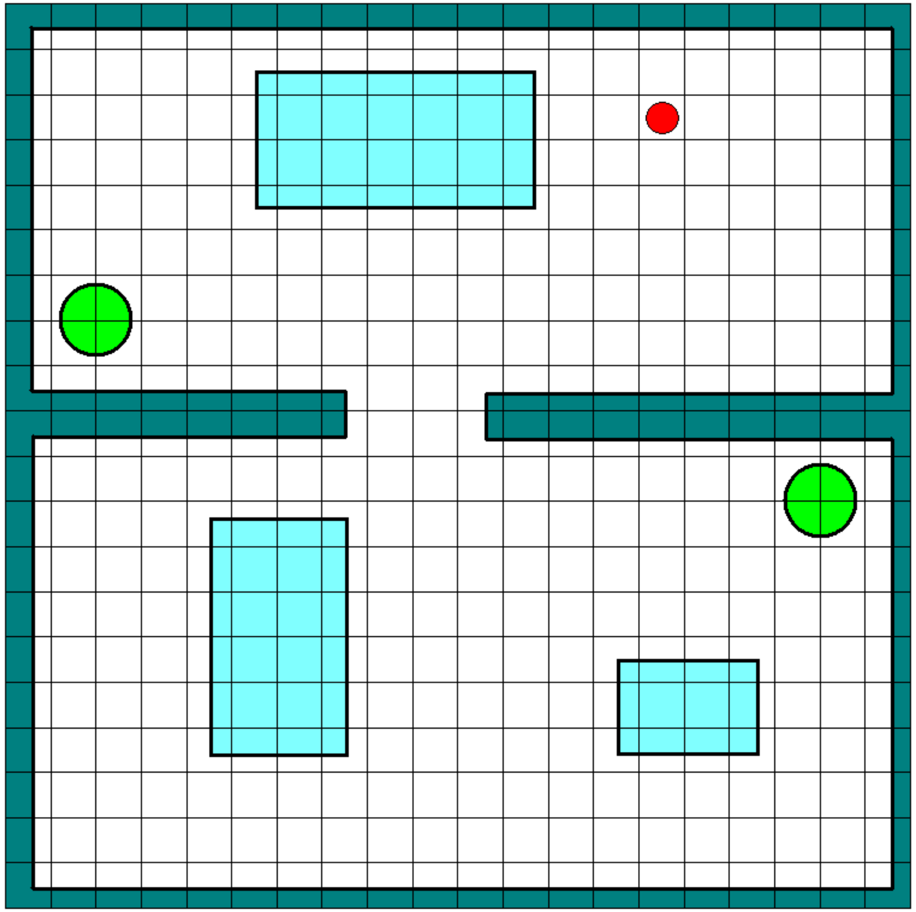


For an Ackerman steer (car-like) drive

Exploration Subsystem

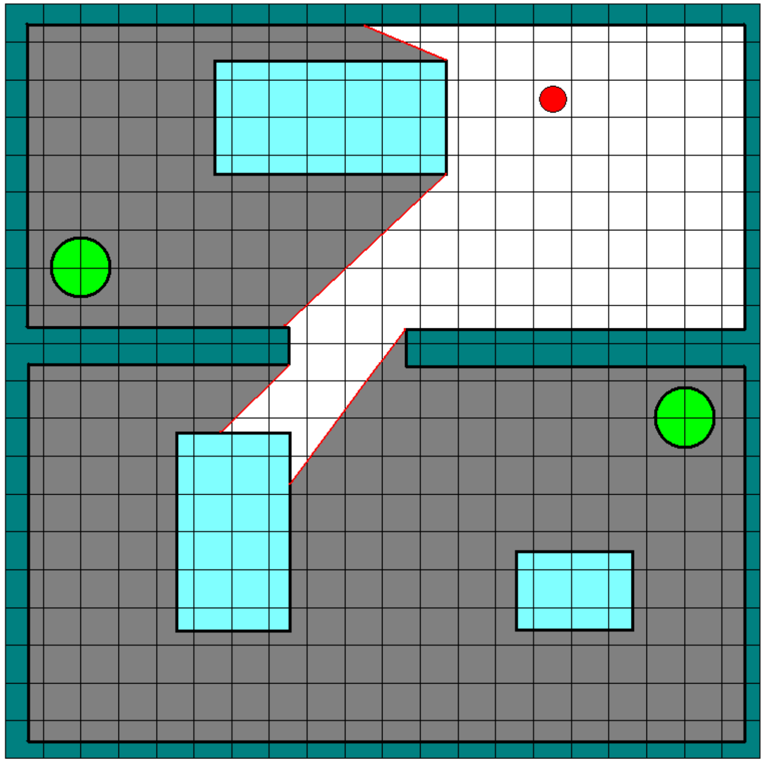
- **Inputs:** The map constructed by the robot so far
- **Output:** Next best location to move to further the process of building the map
- **Various flavors:**
  - *Frontier based approach*: Move to the closest frontier (range finder based sensing modality)
  - *Entropy based approach*: Move to the location of high entropy to maximize information gain (range finder based sensing modality)
  - *Far feature based approach*: Move towards the farthest features of an image (vision based sensing modality)

Exploration Subsystem: An Example

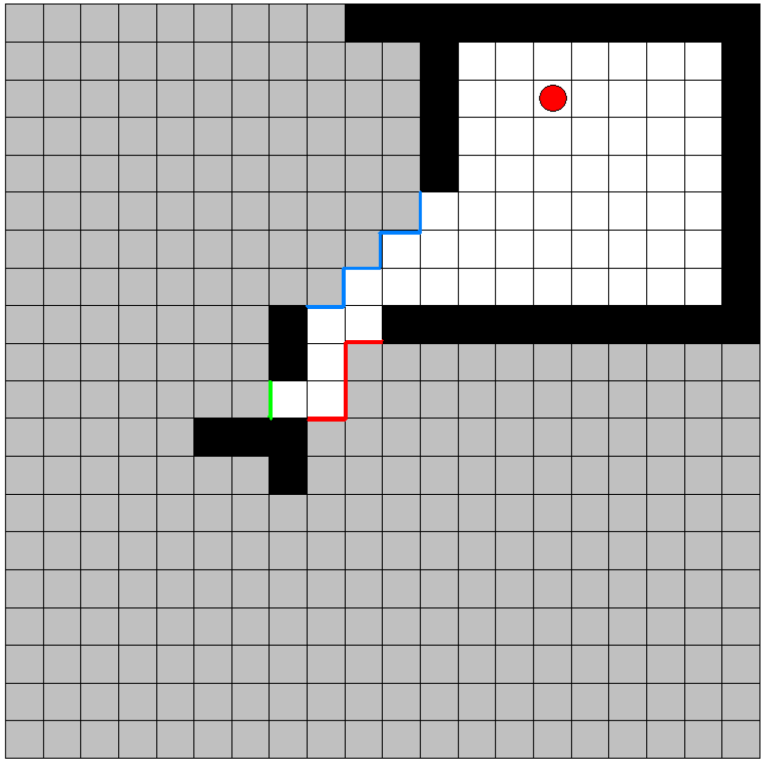


Two-room layout with initial robot position

Example: Scan 1

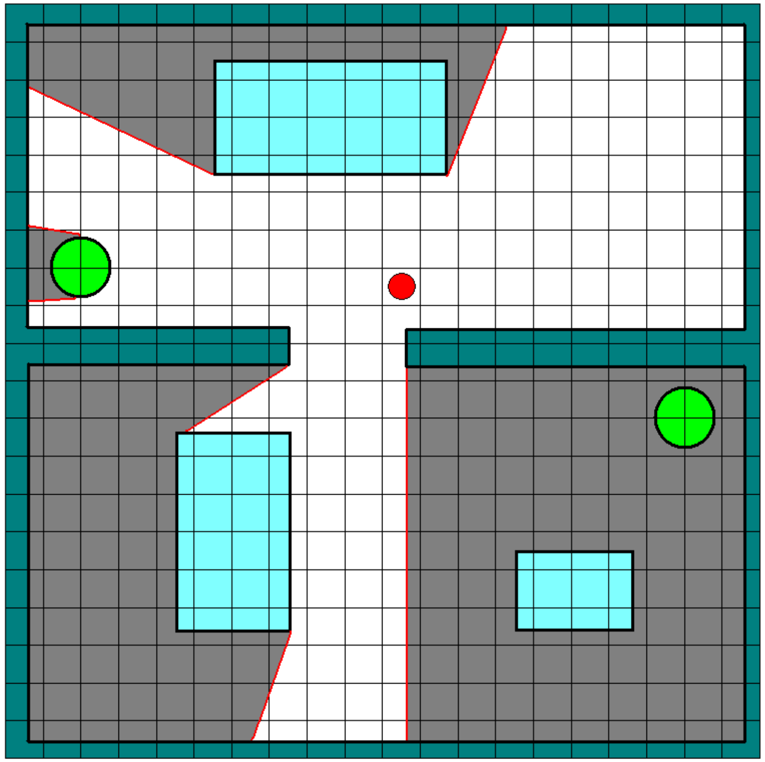


Laser-Limited Sensor Sweep #1

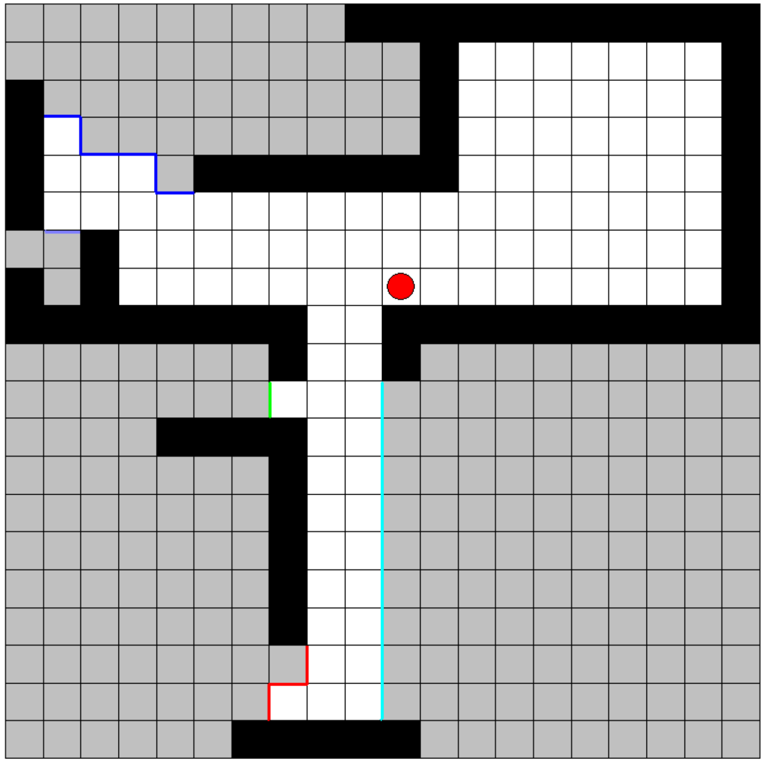


Coverage Grid – Initial Build

Example: Scan 2

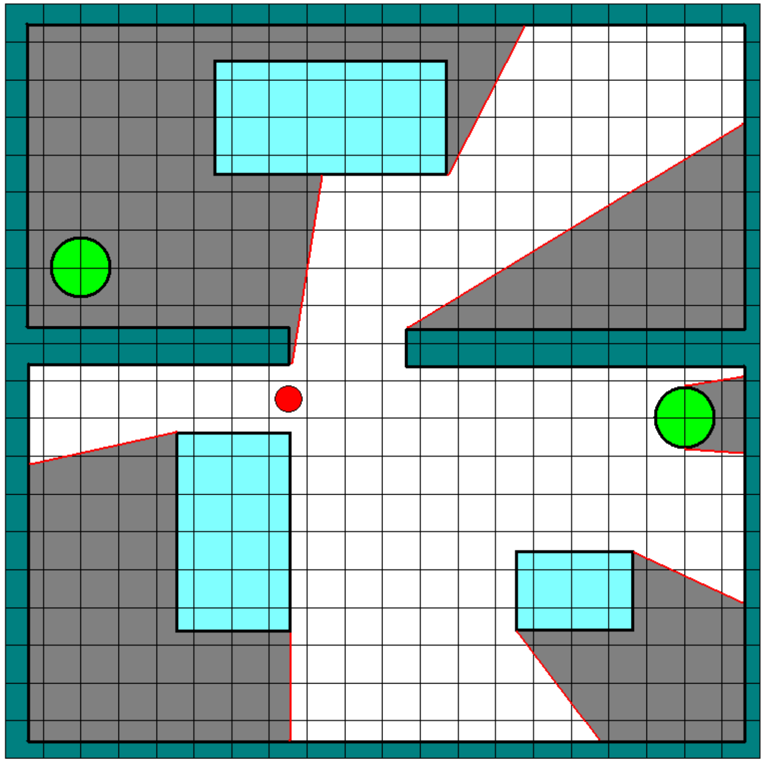


Laser-Limited Sensor Sweep #2

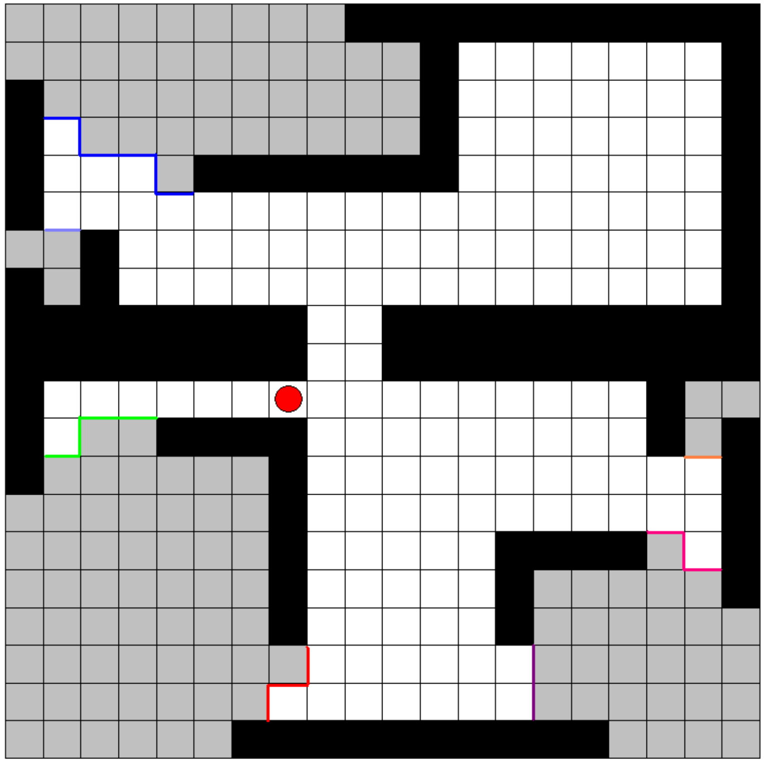


Coverage Grid – Update #2

Example: Scan 3

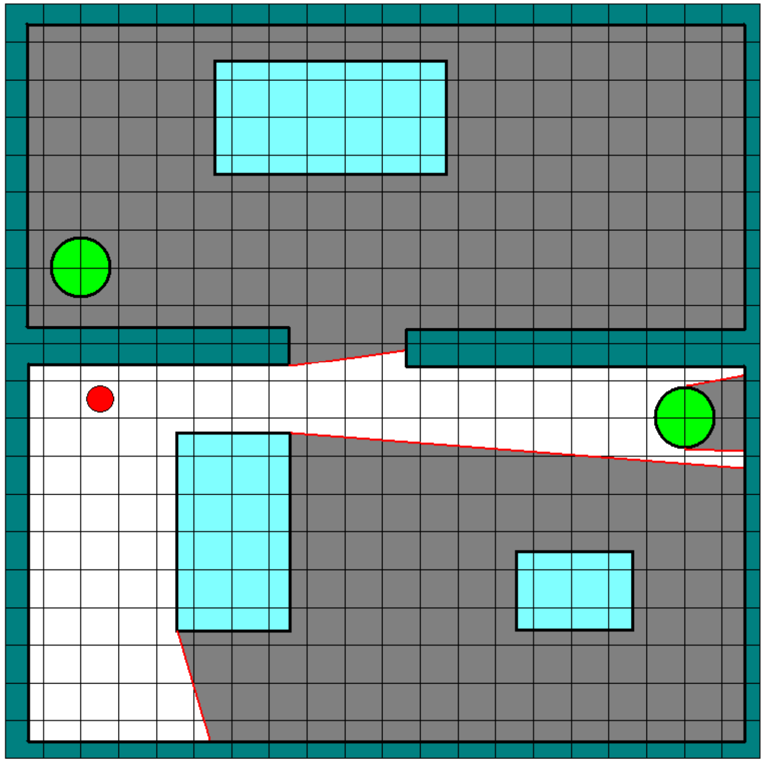


Laser-Limited Sensor Sweep #3

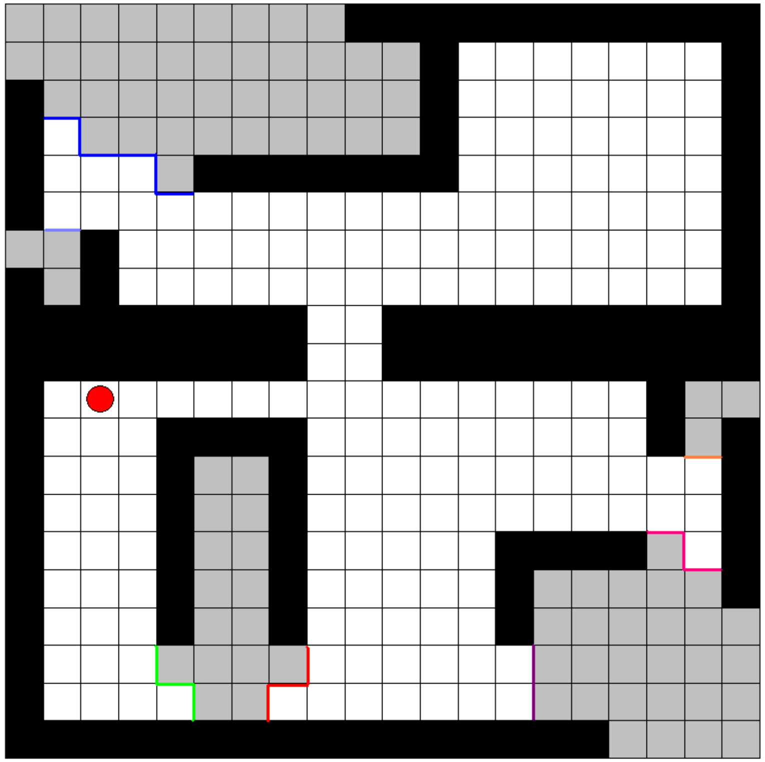


Coverage Grid – Update #3

Example: Scan 4

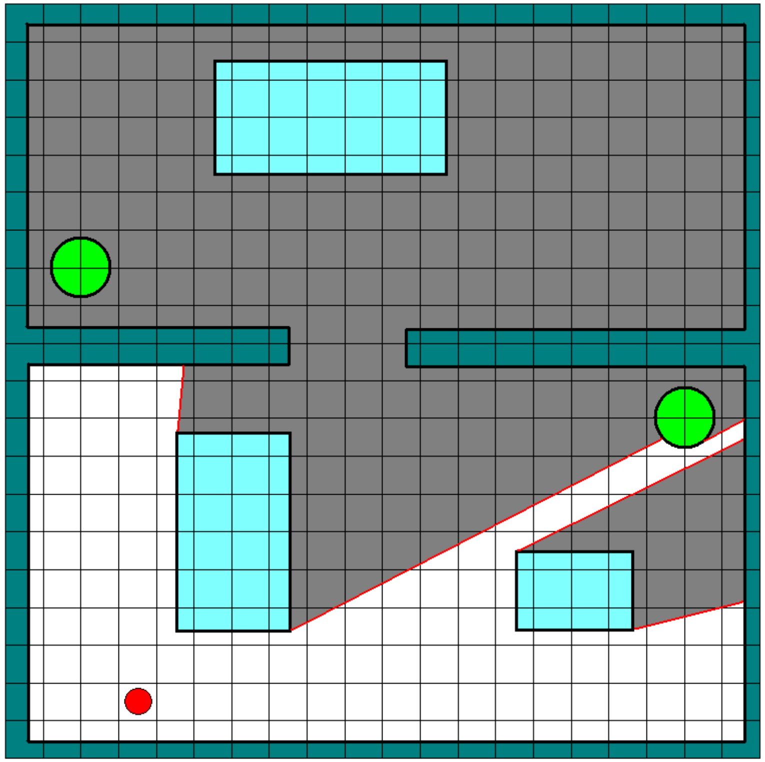


Laser-Limited Sensor Sweep #4

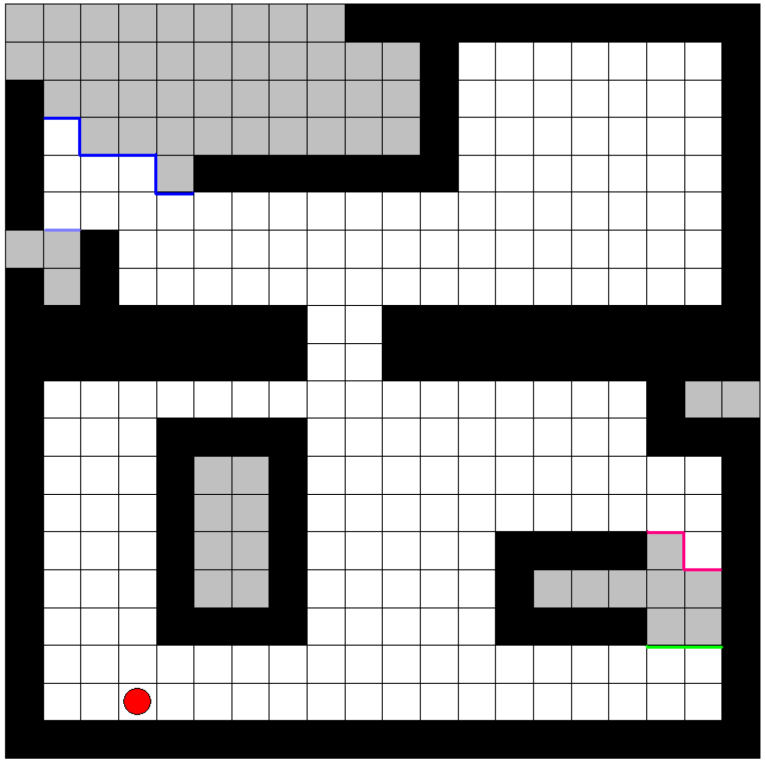


Coverage Grid – Update #4

Example: Scan 5

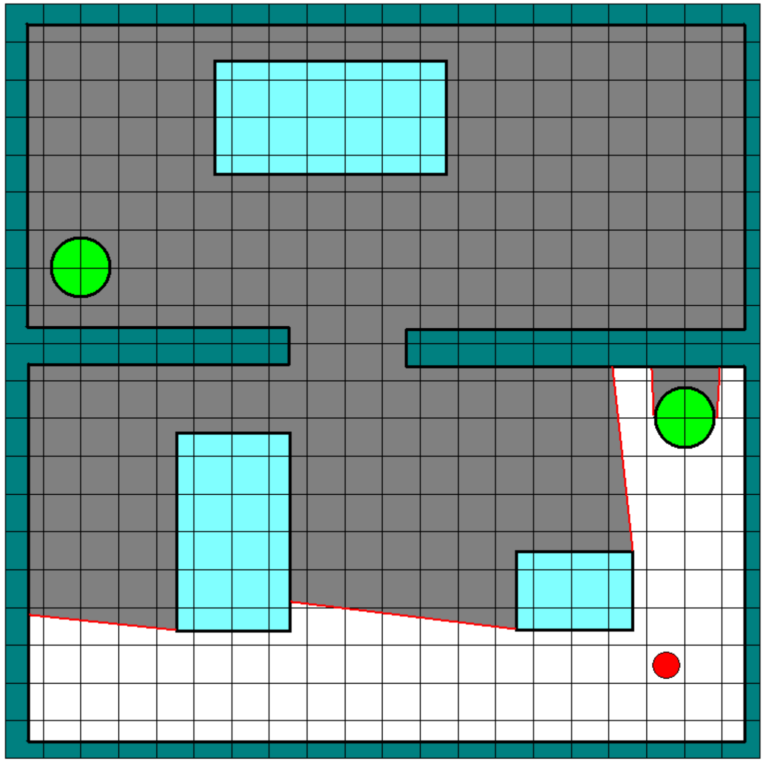


Laser-Limited Sensor Sweep #5

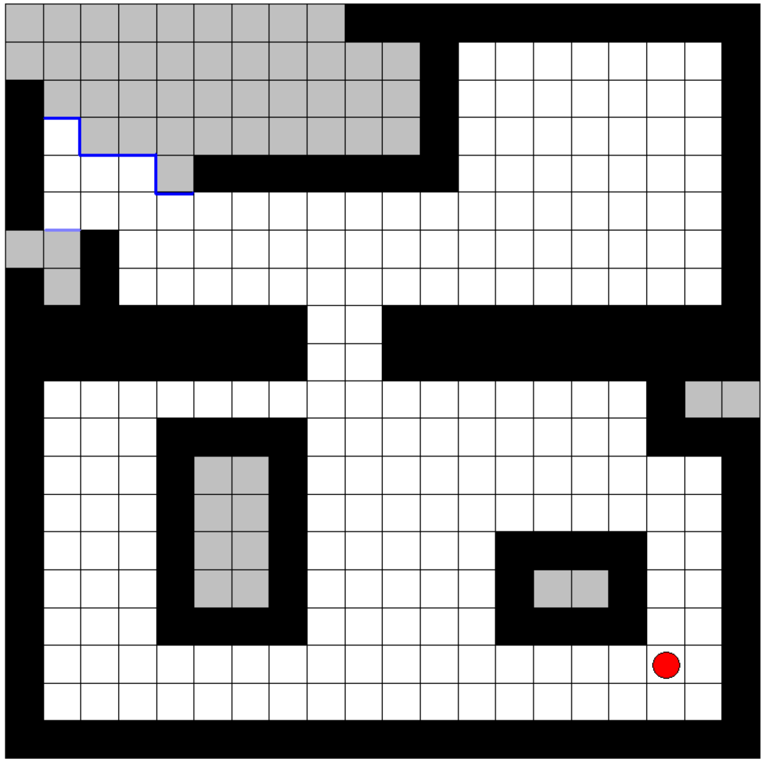


Coverage Grid – Update #5

Example: Scan 6

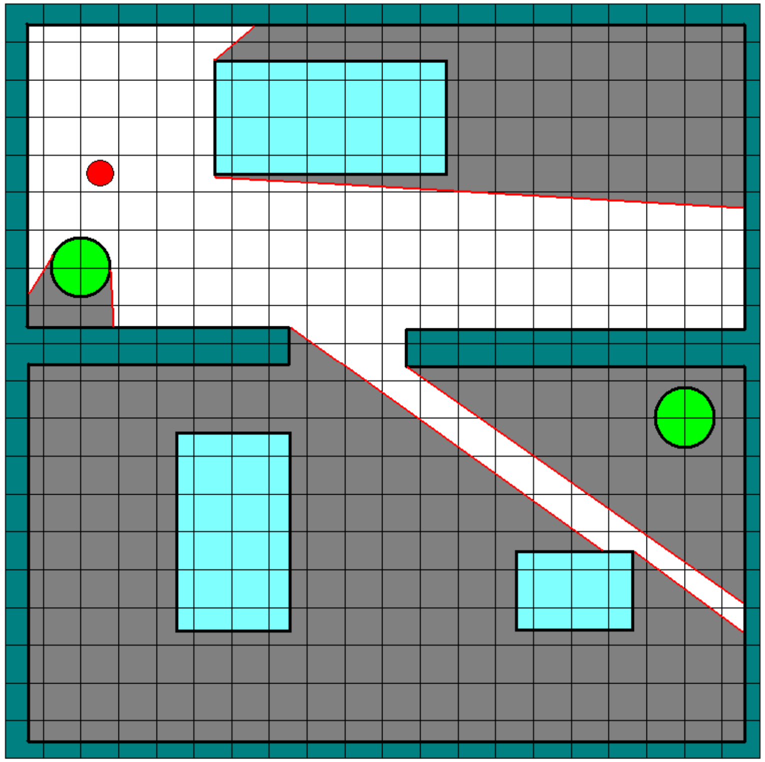


Laser-Limited Sensor
Sweep #6

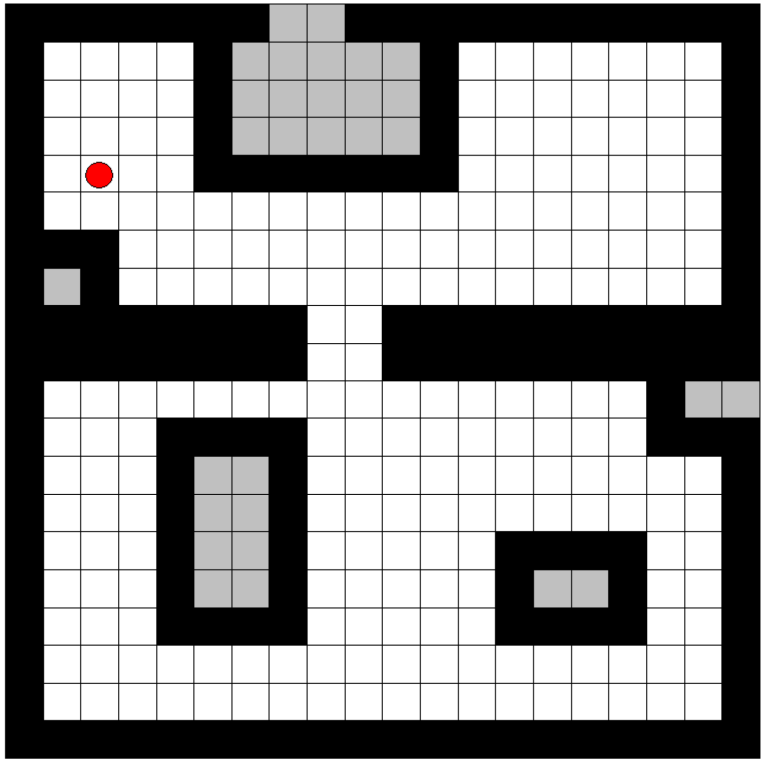


Coverage Grid – Update
#6

Example: Scan 7



Laser-Limited Sensor Sweep #7



Coverage Grid – Update #7

Exploration complete – all accessible frontiers have been visited.

Reactive Navigation Subsystem

Reactive Navigation Strategy	Description
Potential fields	Given a start and goal pose in an unknown world model the obstacles sensed in the most recent scan by repulsive potentials or forces and target by attractive forces. The robot moves in the direction of the resultant force
Fuzzy Logic Based	Given a start and goal pose in an unknown world and the most recent sensor scan, model obstacle avoidance behavior and goal orienting behavior through fuzzy rule –base. Combine behaviors through another set of fuzzy rules
Nearness Diagram	In this method we divide the surroundings into sectors and then find out the nearness of each of these sectors. Based on the information received a RND is constructed depicting the nearness factor of various obstacles. This diagram is used to find out the closest gap to goal and guide the robot towards the goal avoiding the obstacles

Applications

Search and rescue

Rough Terrain Navigation

(b)

Outdoor Autonomous Navigation

Some Applications

TOURBOT: interactive tour-guide robot able to provide individual access to museums' exhibits and cultural heritage both over the internet and on-site in museum

http://www.ics.forth.gr/tourbot/

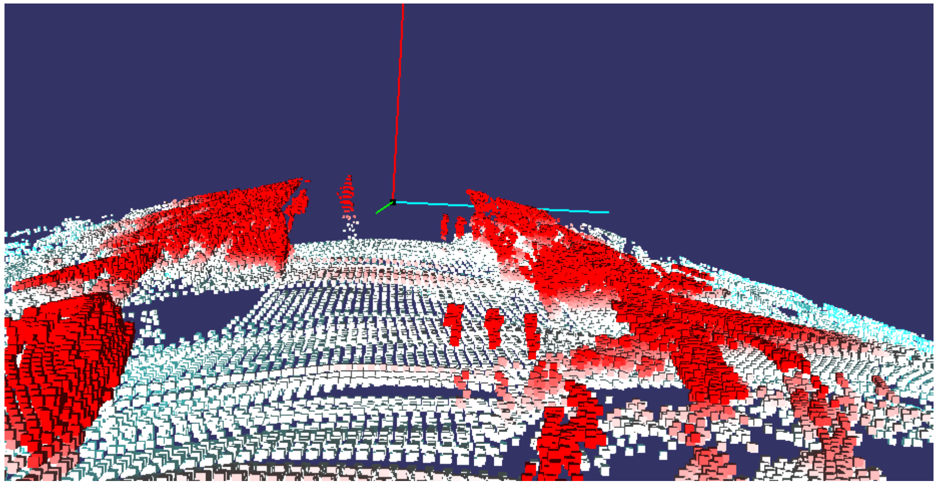
NURSEBOT: Robotic assistants for the elderly.

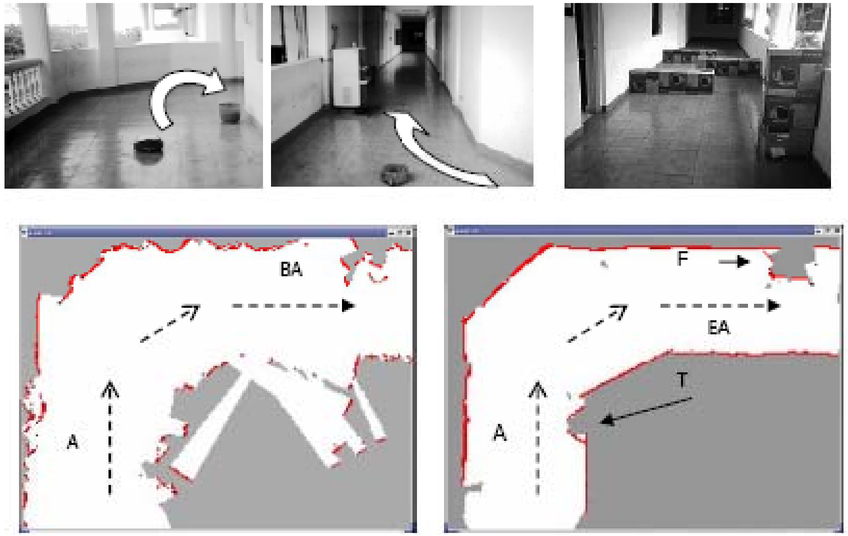
http://www.ri.cmu.edu/projects/project_347.html

SENSING ROBOT (Silent Commerce): For wear-house inspection to prevent thefts and losses

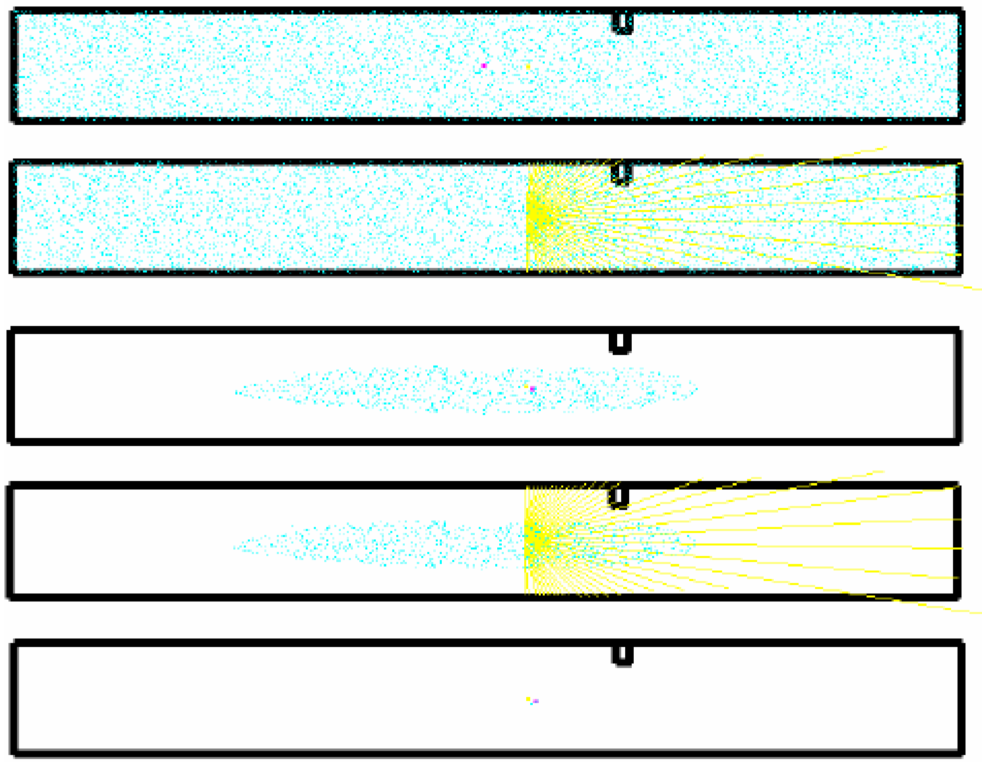
http://www.accenture.com/xd/xd.asp?it=enweb&xd=services\technology\vision\tech_sensing_robot.xml

Mapping





LOCALIZATION



REACTIVE NAVIGATION