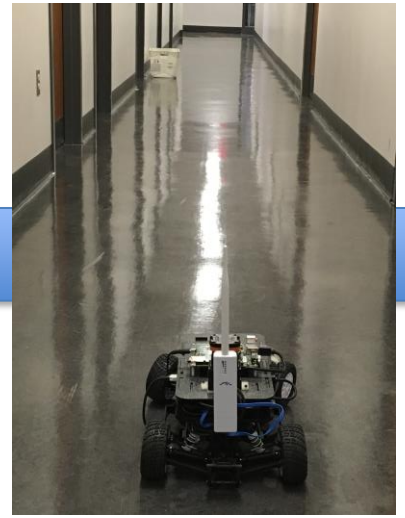
F1/10th Autonomous Racing

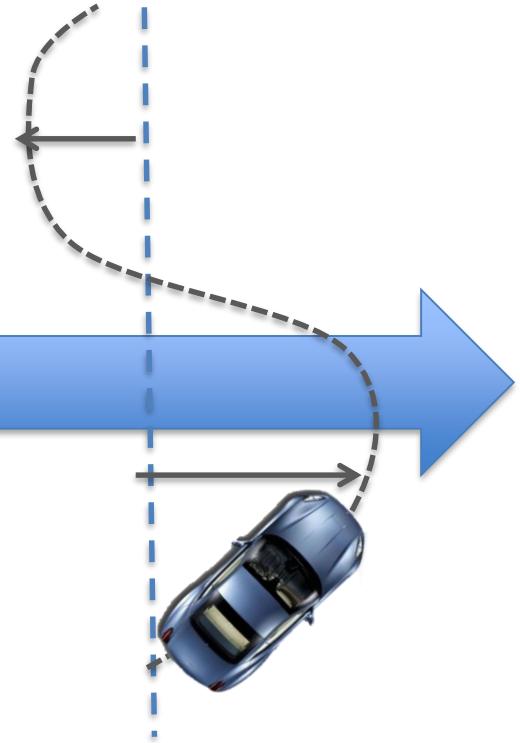
Simultaneous Localization & Mapping

Paril Jain

Penn Engineering

Previous Week





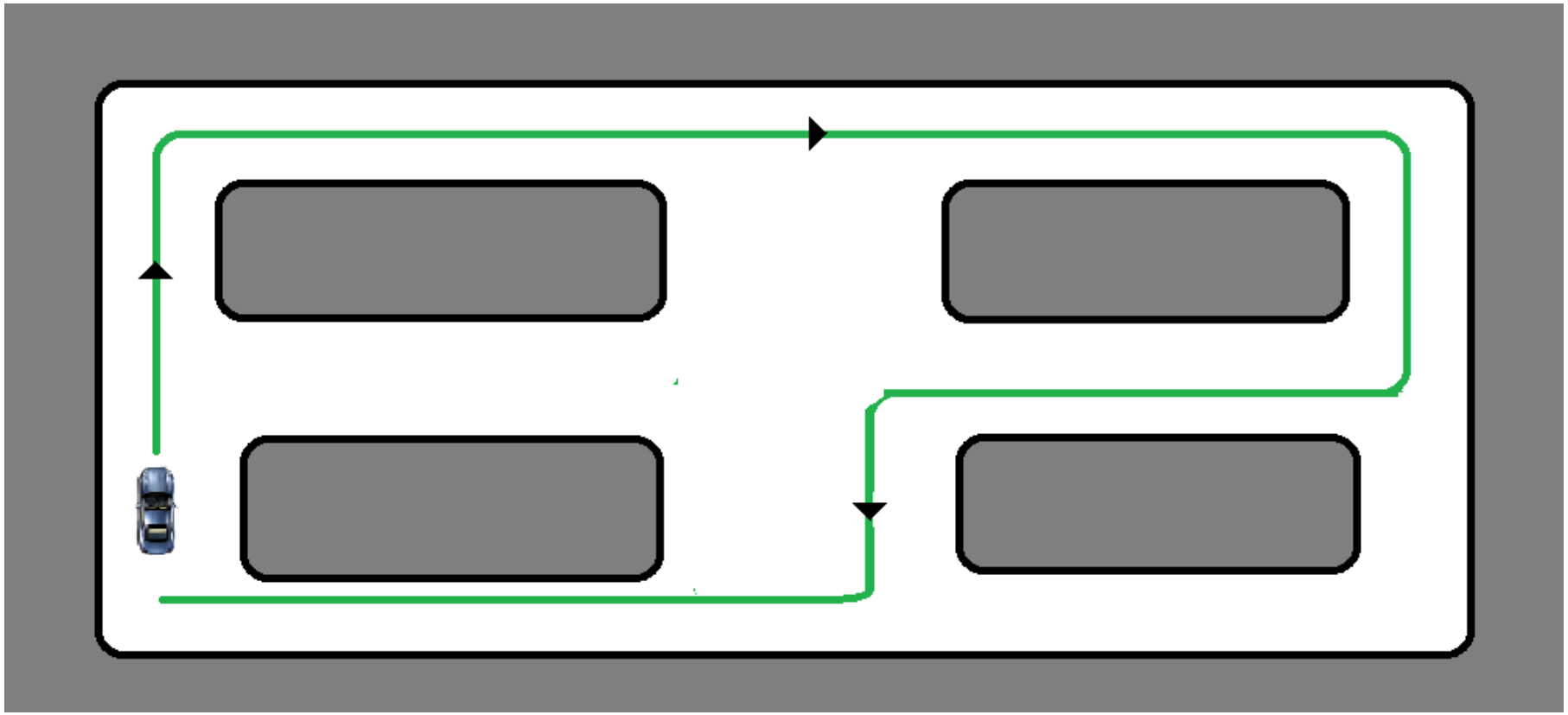
IMU and LIDAR

Localization

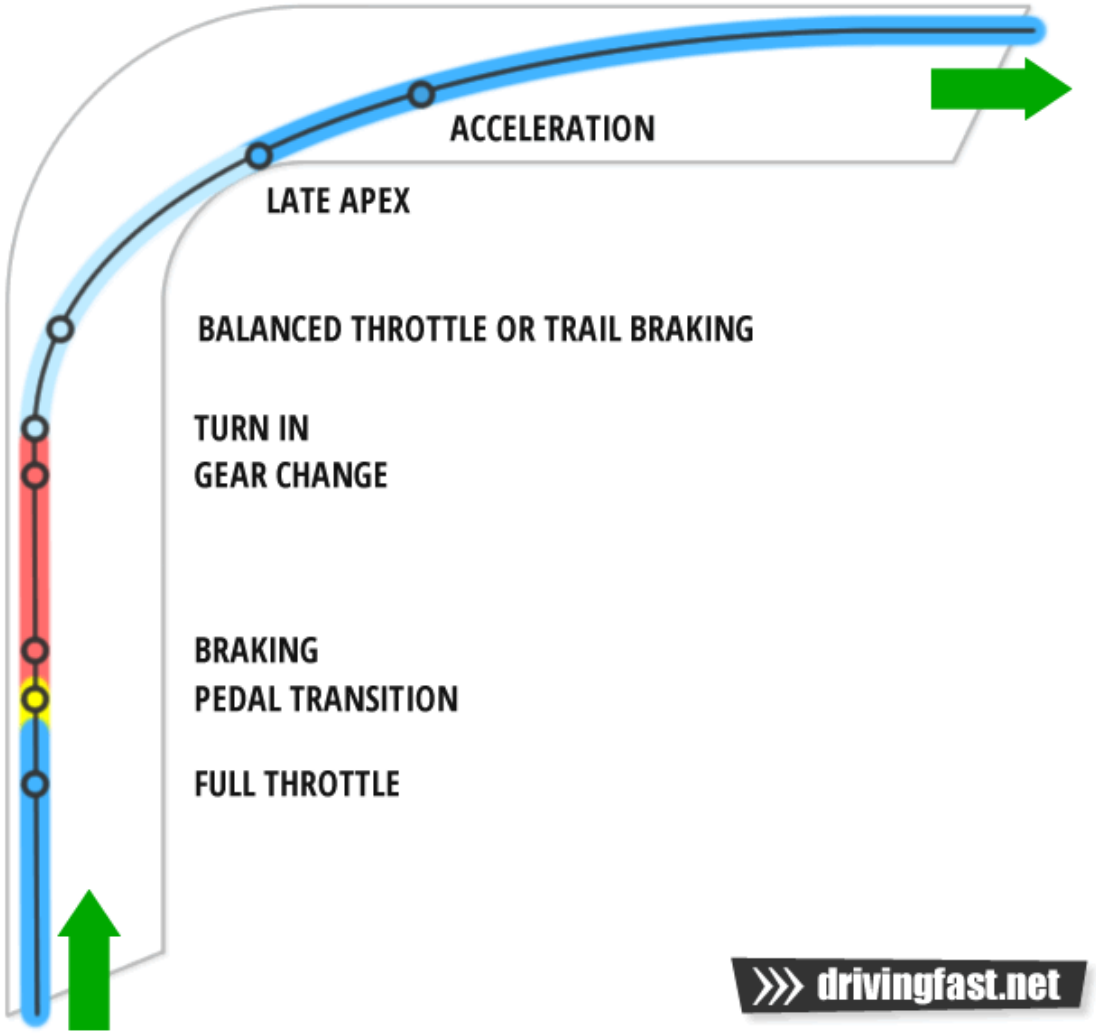
PID Control

Limitations : Basic Path Planning

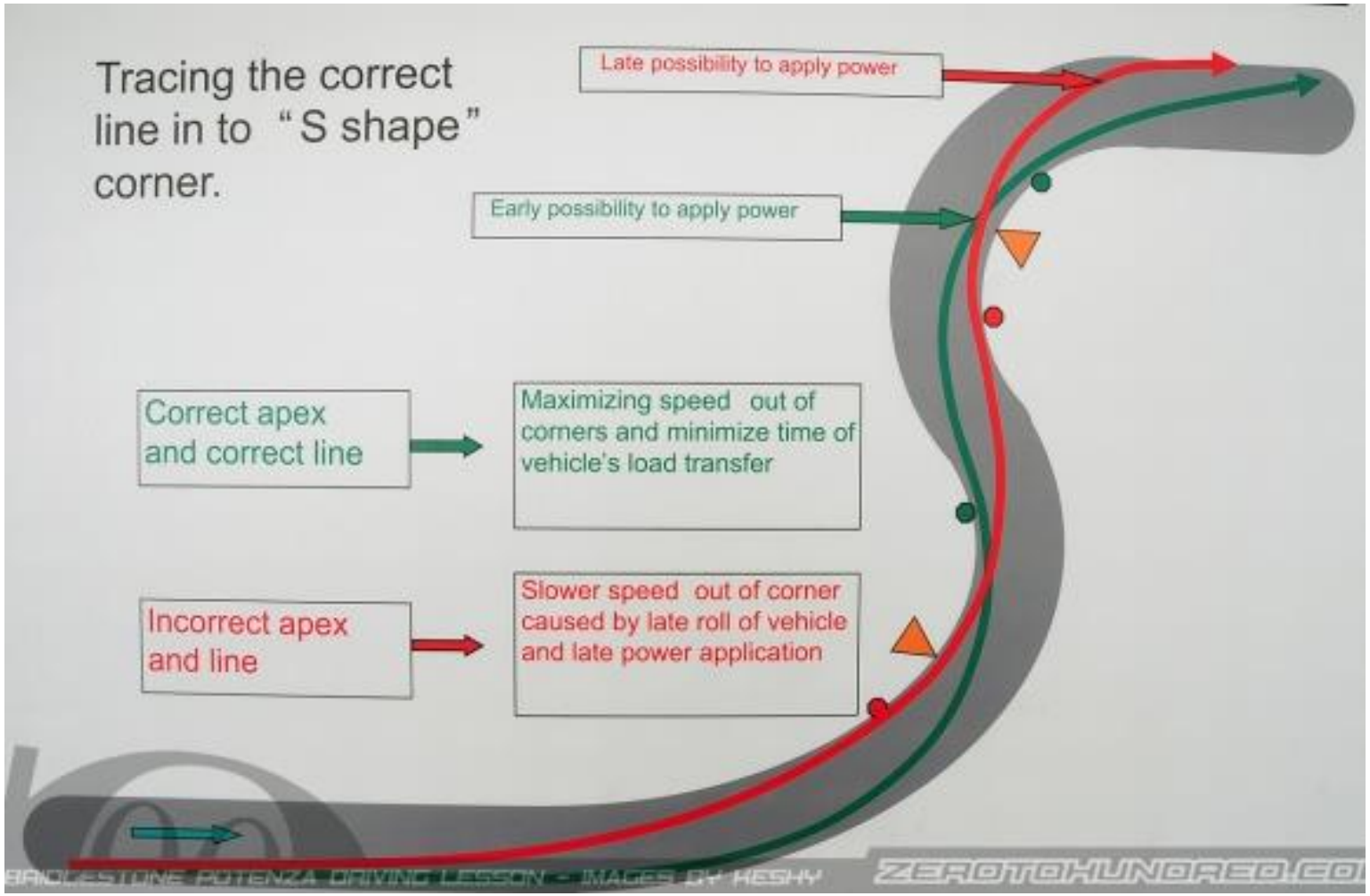
- High Level Path Assignments
  - 2nd right, 2nd right, 1st right, 1st left, 1st right



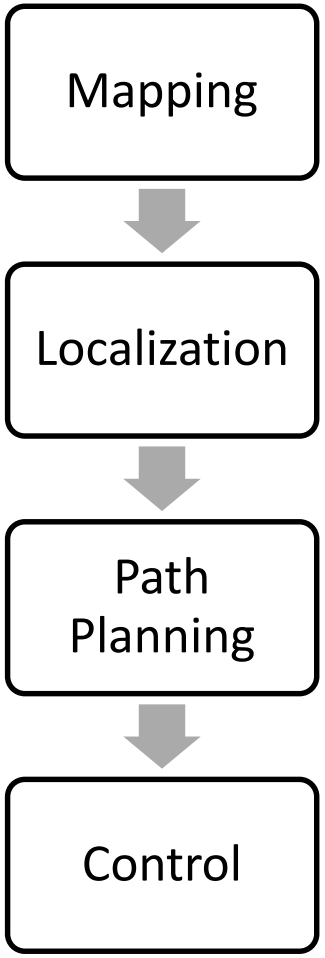
Race Lines


ACCELERATION
LATE APEX
BALANCED THROTTLE OR TRAIL BRAKING
TURN IN
GEAR CHANGE
BRAKING
PEDAL TRANSITION
FULL THROTTLE
drivingfast.net

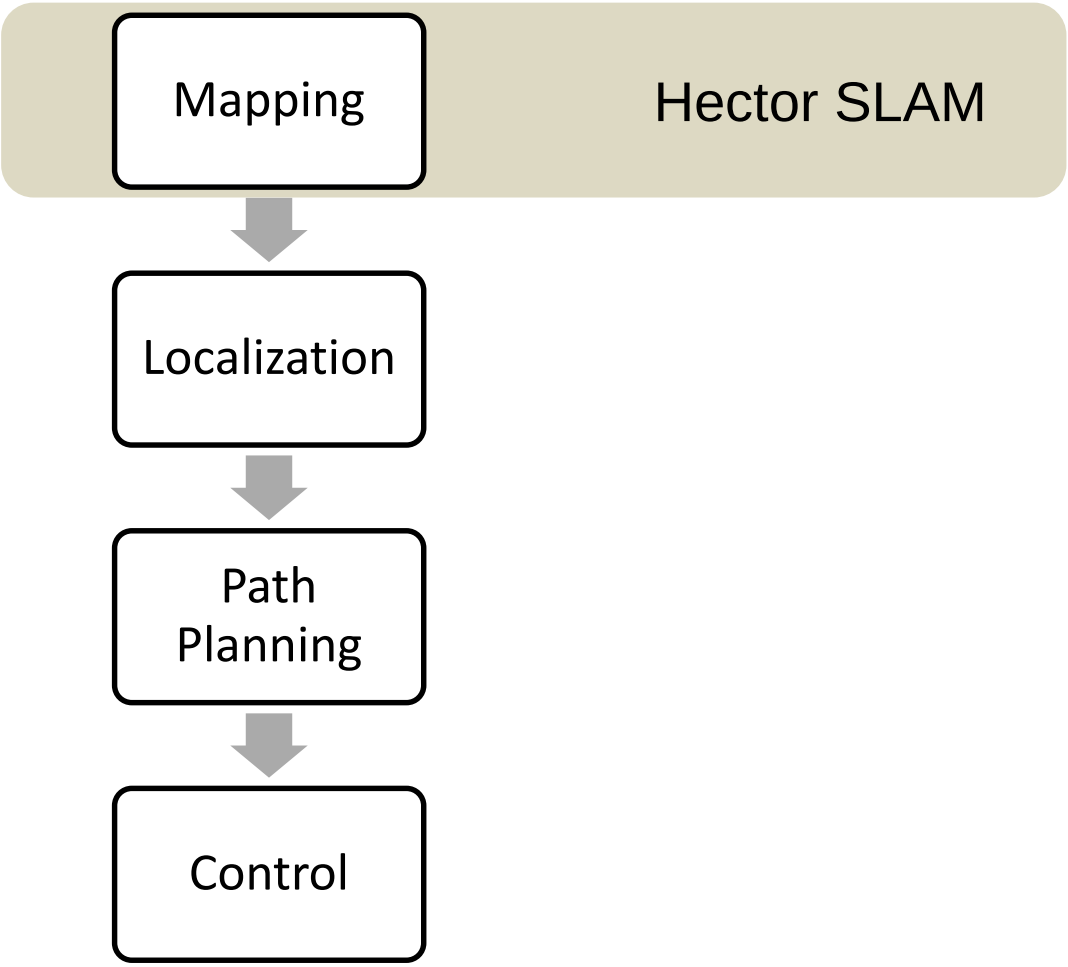
Limitations : No Future Information



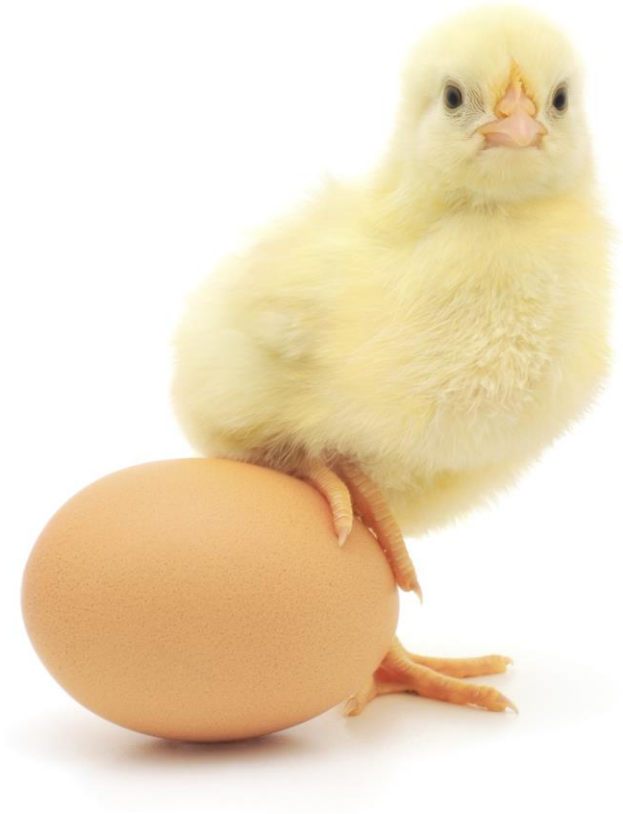
System Overview

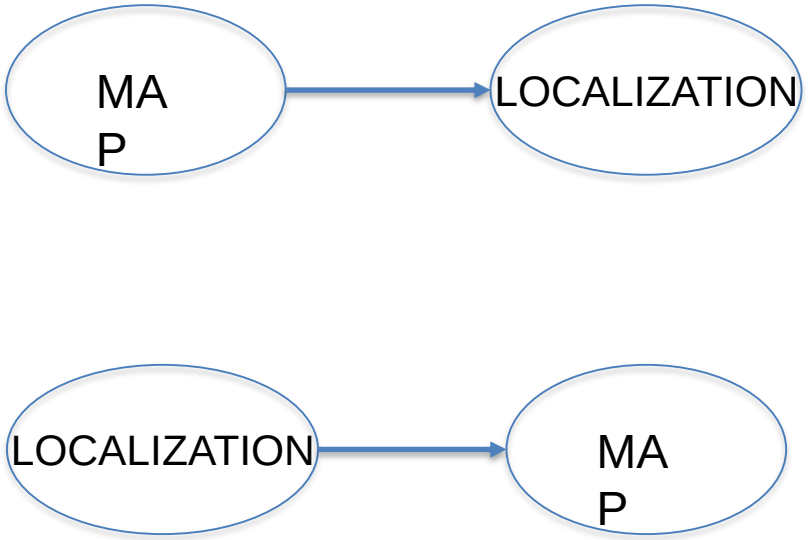


System Overview

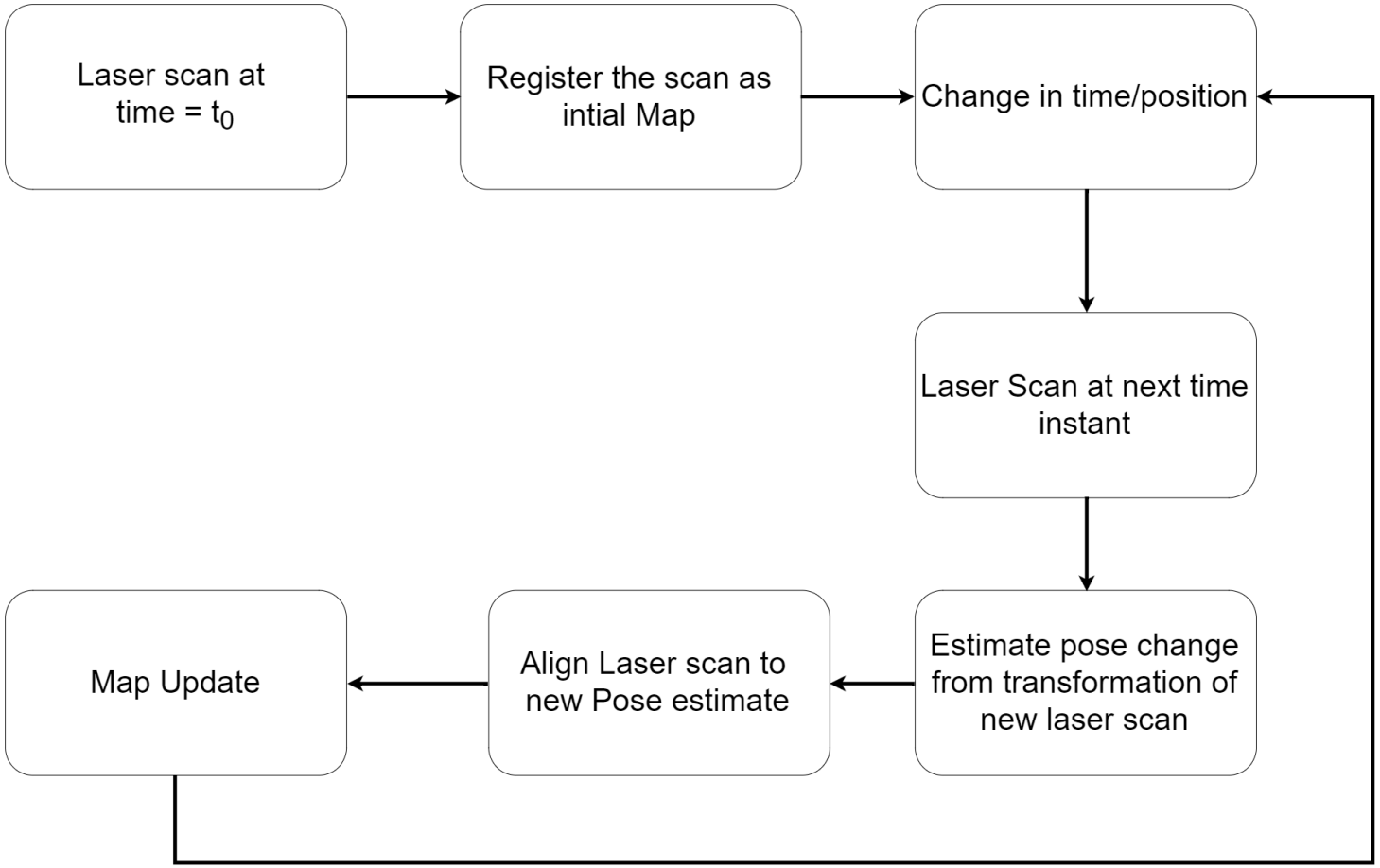


SLAM : A Chicken-Egg problem





Overview of SLAM

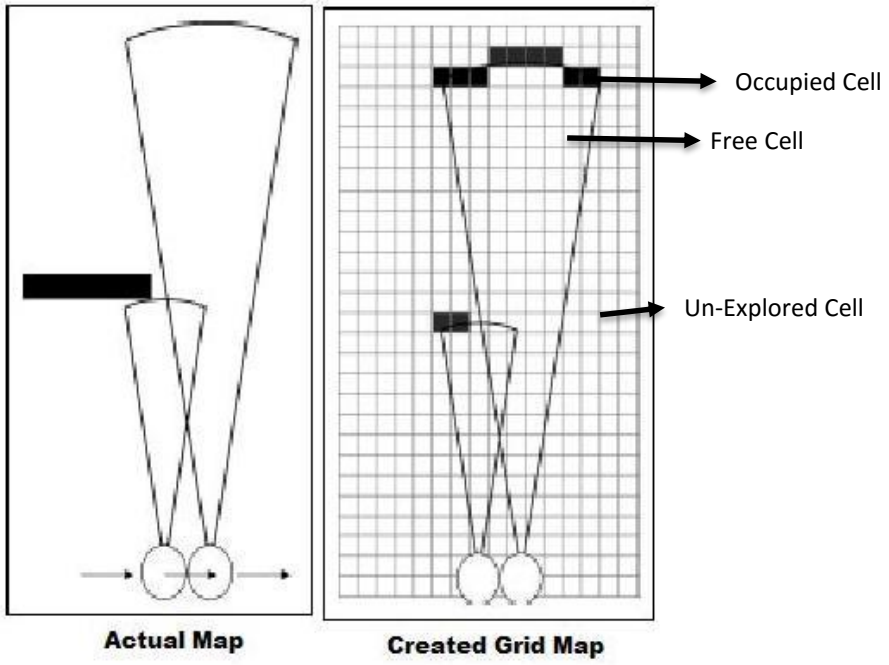


Video provided separately

- Car running in corridor
- Map being generated
- Video will be used for explaining the overview

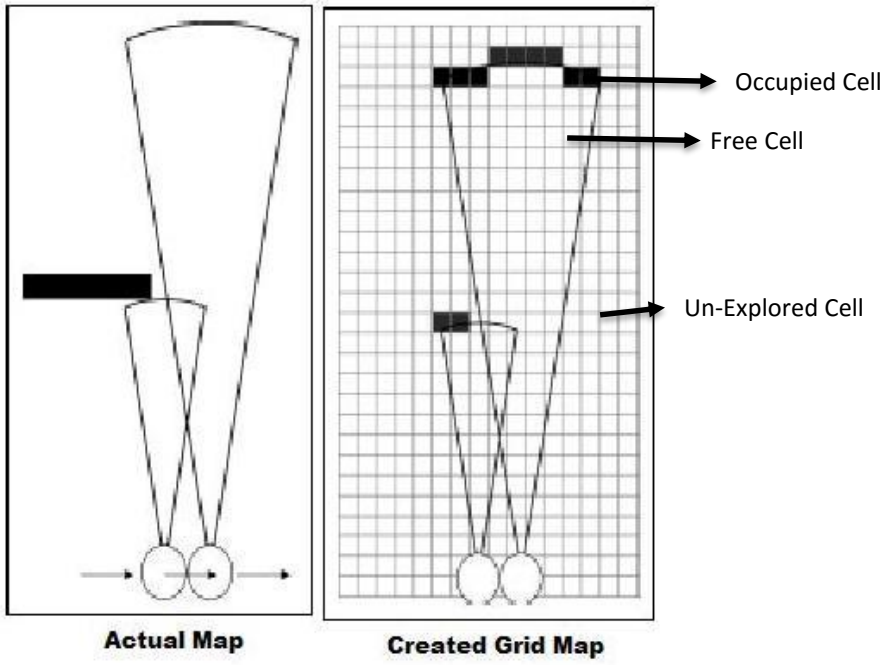
Occupancy Grid Mapping

Measurement Model



Occupancy Grid Mapping

Measurement Model



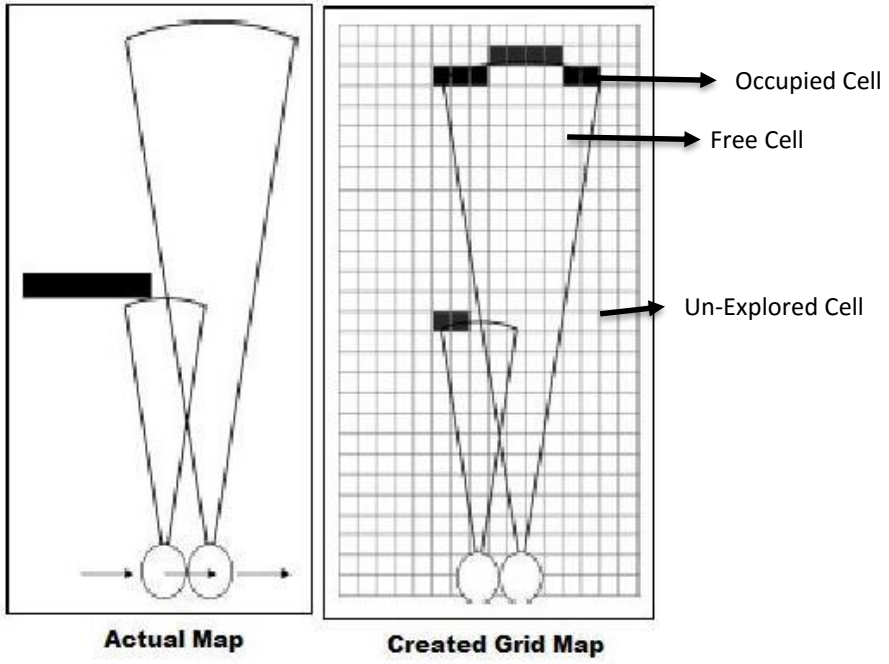
- Measurement :

  $m_{x,y} = 1$ LiDAR hit

  $m_{x,y} = 0$ No occlusion

Occupancy Grid Mapping

Measurement Model



- Measurement :

  $m_{x,y} = 1$ LiDAR hit

  $m_{x,y} = 0$ No occlusion

- Map Cell:

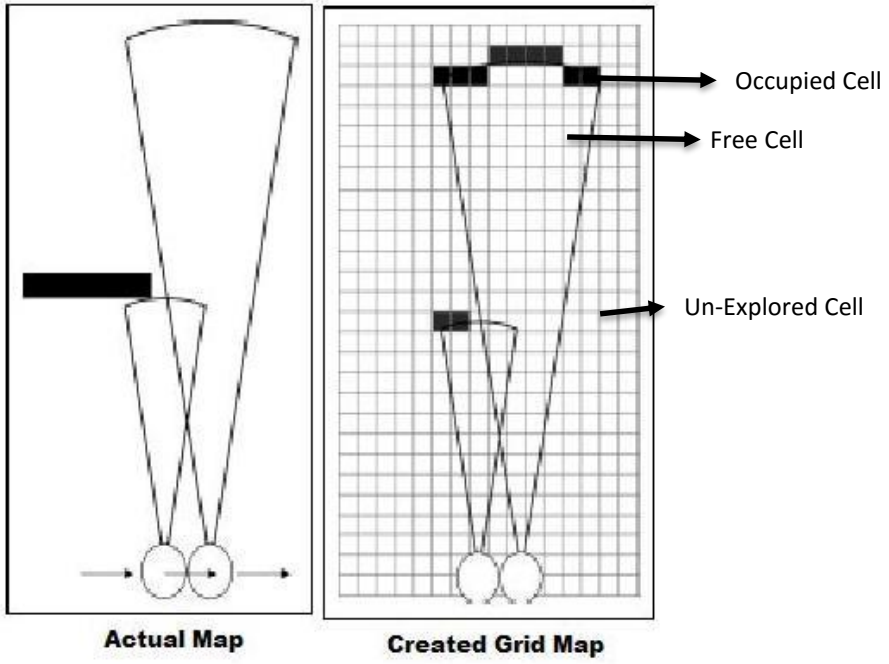
  $Z = 1$ Occupied

  $Z = 0$ UnExplored

  $Z = -1$ Free

Occupancy Grid Mapping

Measurement Model



- Measurement :

  $m_{x,y} = 1$ LiDAR hit

  $m_{x,y} = 0$ No occlusion

- Map Cell:

  $Z = 1$ Occupied

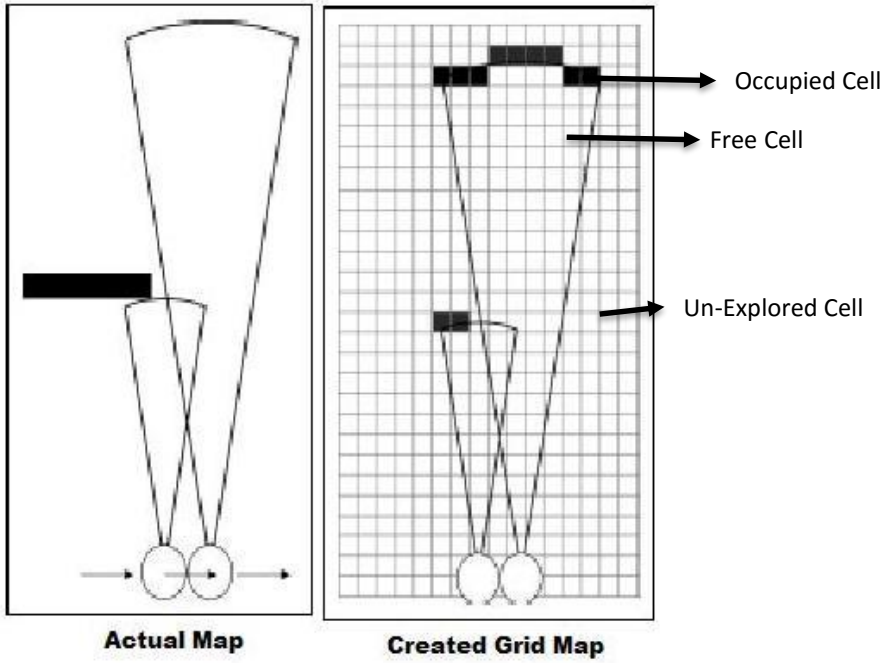
  $Z = 0$ UnExplored

  $Z = -1$ Free

- Measurement Model :

  $p(z|m_{x,y})$

Occupancy Grid Mapping



$$\log odd\_occ := \log \frac{p(z=1|m_{x,y}=1)}{p(z=1|m_{x,y}=0)}$$

Log Probability for occupied cells

$$\log odd\_free := \log \frac{p(z=-1|m_{x,y}=0)}{p(z=-1|m_{x,y}=1)}$$

Log Probability for free cells

Occupancy Grid Mapping

Map Update :

- Cells with z = 1:
  - log `odd` = log `odd` + log `odd_occ`
- Cells with z = -1:
  - log `odd` = log `odd` - log `odd_free`

Occupancy Grid Mapping

Map Update :

- Cells with z = 1:
  - log `odd` = log `odd` + log `odd_occ`
- Cells with z = -1:
  - log `odd` = log `odd` - log `odd_free`

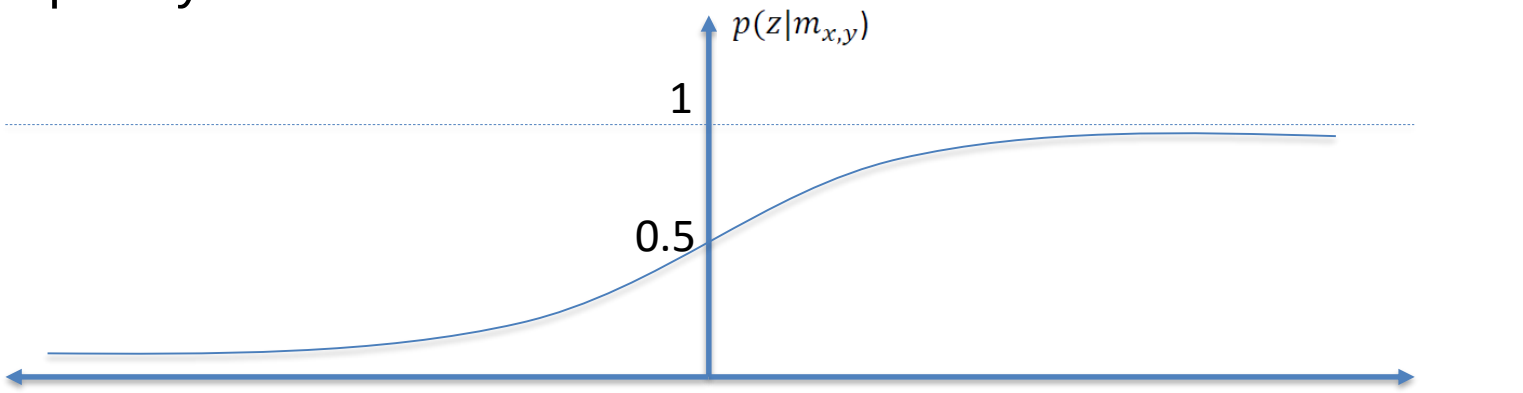
- Threshold the cell values by upper /lowerlimit to avoid being completely certain

Occupancy Grid Mapping

Map Update :

- Cells with z = 1:
  - log `odd` = log `odd` + log `odd_occ`
- Cells with z = -1:
  - log `odd` = log `odd` - log `odd_free`

- Threshold the cell values by upper /lowerlimit to avoid being completely certain

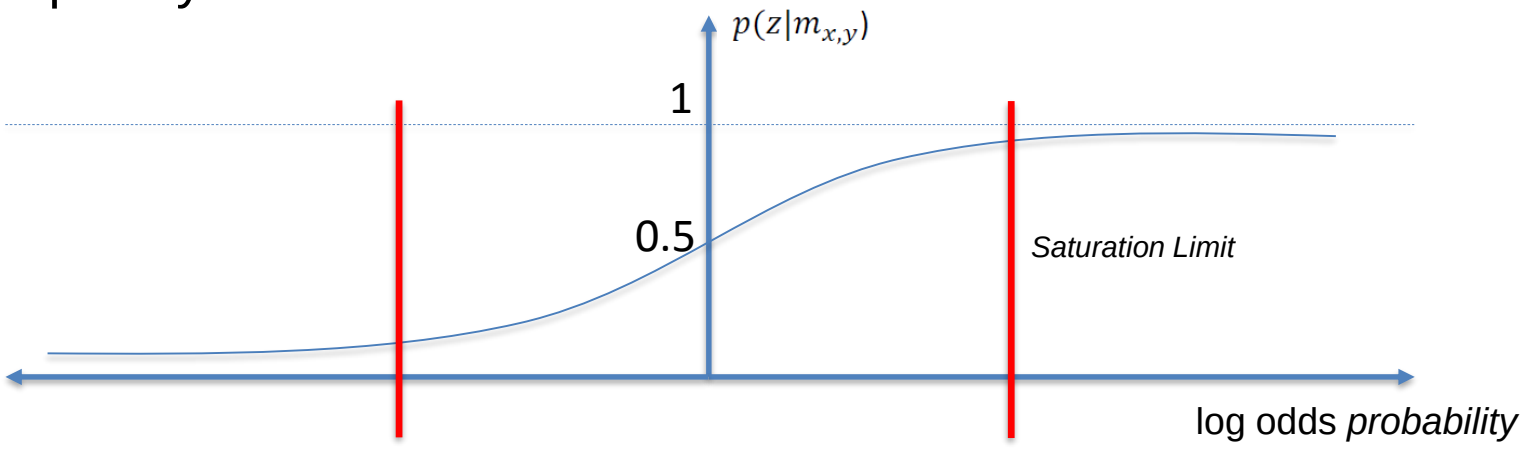


Occupancy Grid Mapping

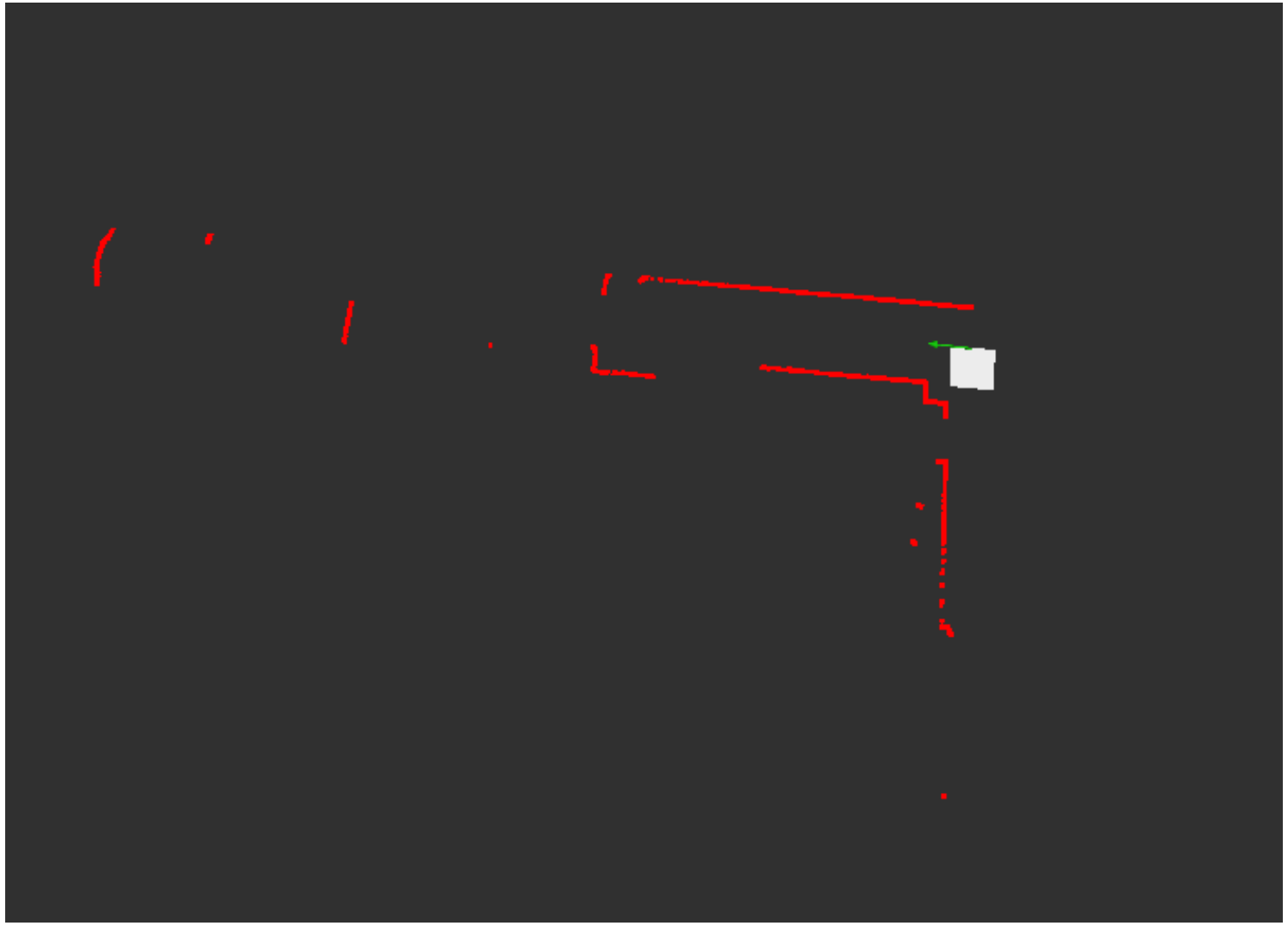
Map Update :

- Cells with z = 1:
  - log `odd` = log `odd` + log `odd_occ`
- Cells with z = -1:
  - log `odd` = log `odd` - log `odd_free`

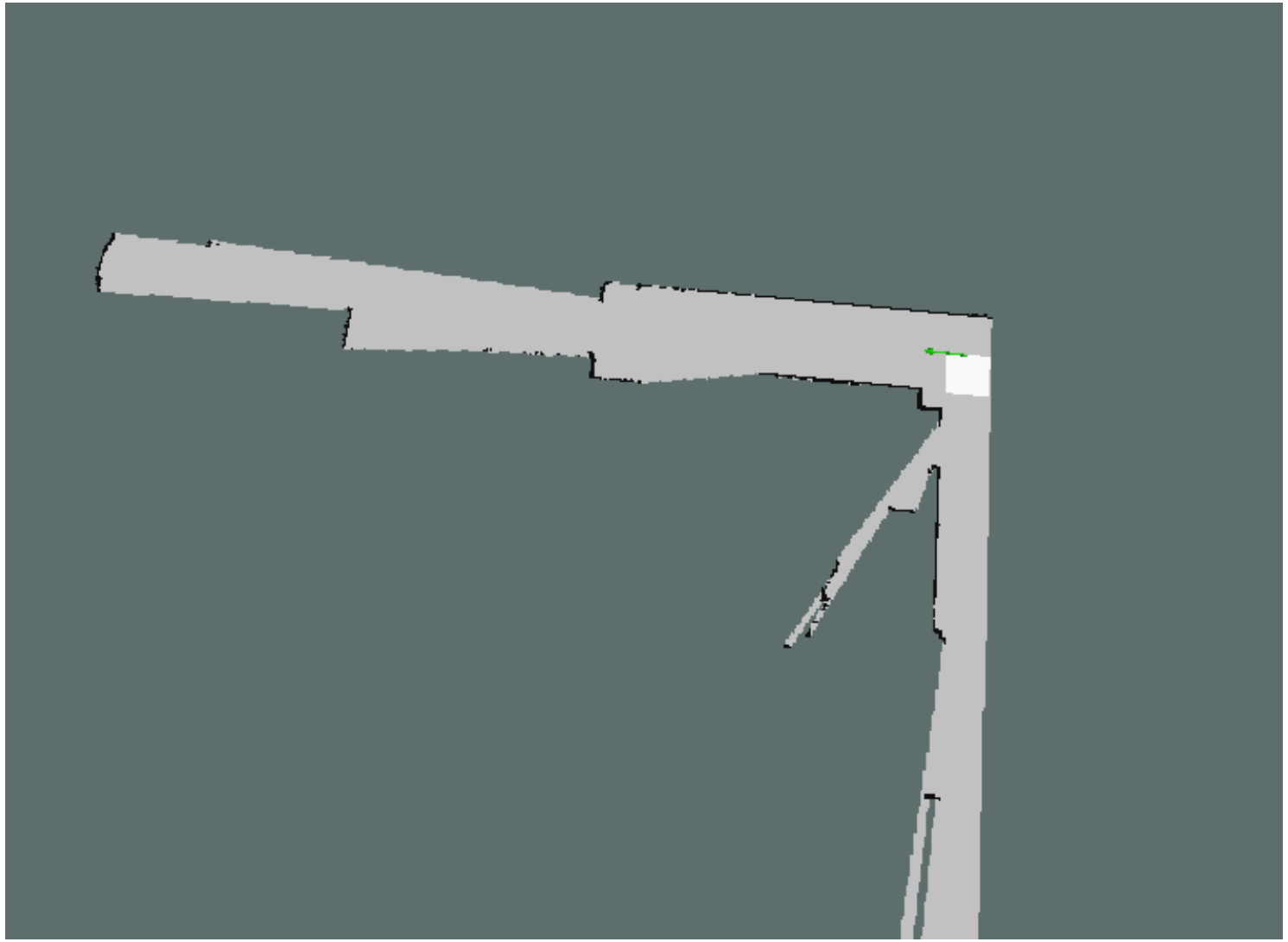
- Threshold the cell values by upper /lowerlimit to avoid being completely certain



Registering the first Scan

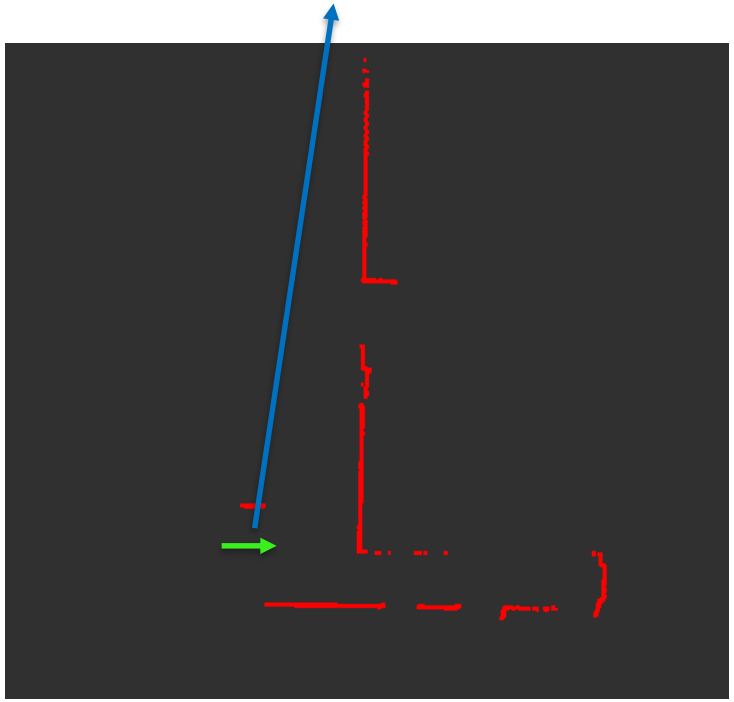


Registering the first Scan



Scan Matching

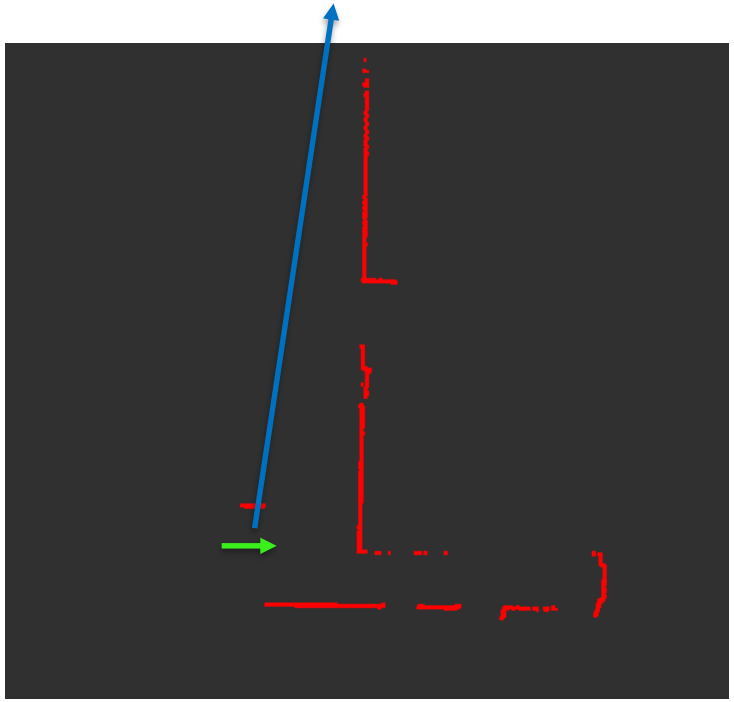
Pose of the Car at t = t1



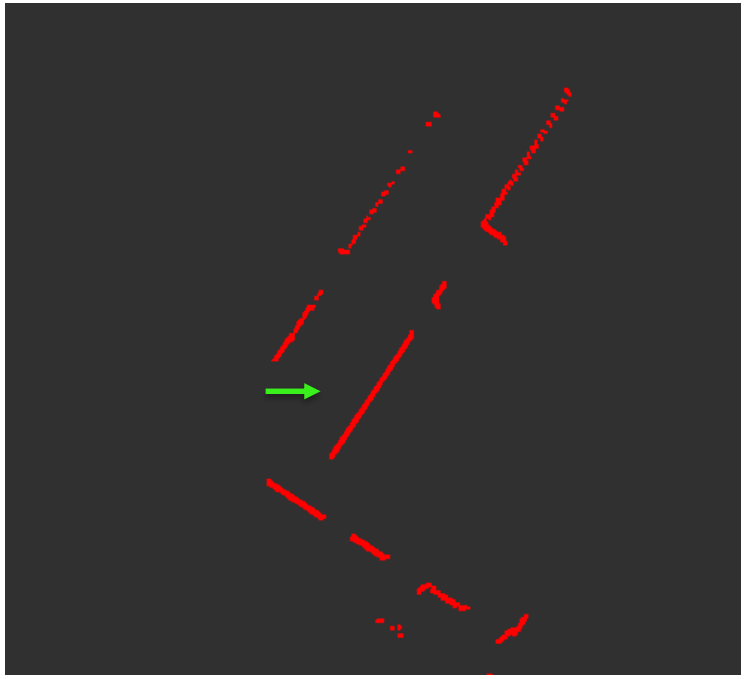
Laser Scans w.r.t car at Time t = t_1

Scan Matching

Pose of the Car at t = t1



Laser Scans w.r.t car at Time t = t_1

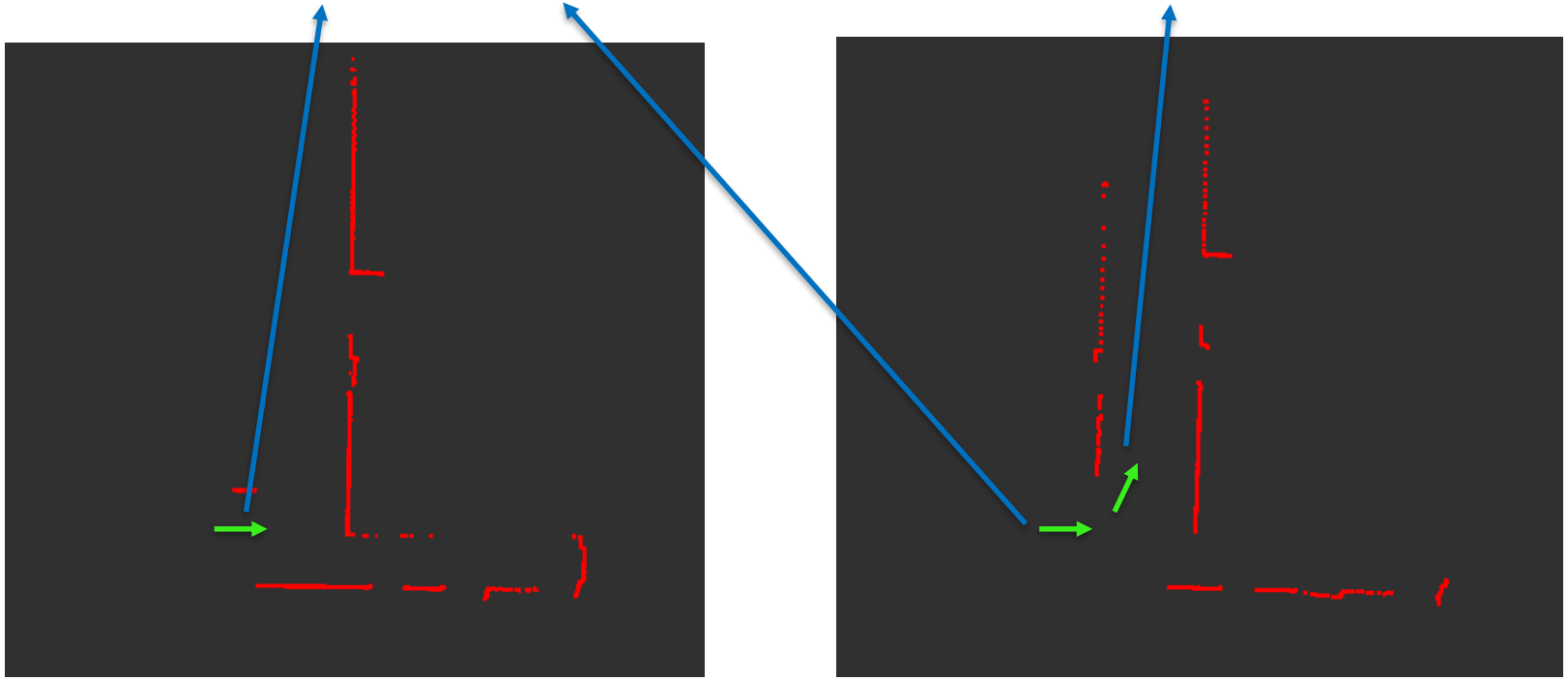


Laser Scans w.r.t car at Time t = t_2

Scan Matching

Pose of the Car at t = t1

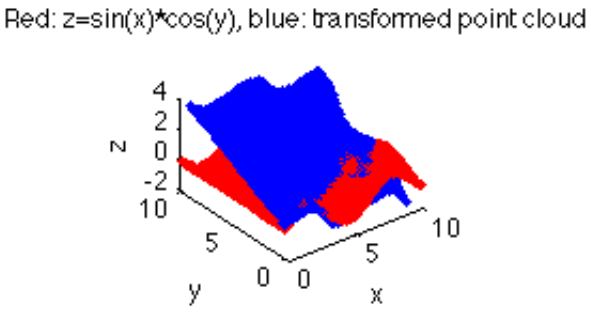
Pose of the Car at t=t2



Laser Scans w.r.t car at Time t = t_1

Laser Scans w.r.t car at Time t = t_2

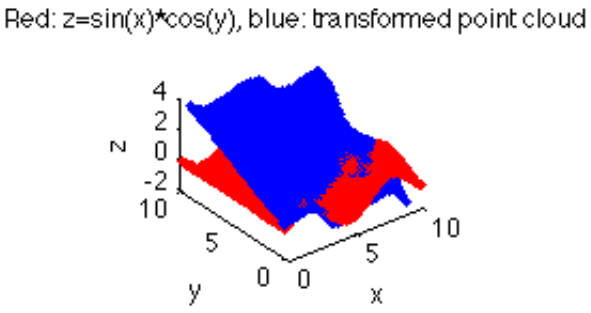
Scan Matching



Iterative Closest Point

Source : Mathworks – File Exchange : Iterative Closest Point Package

Scan Matching

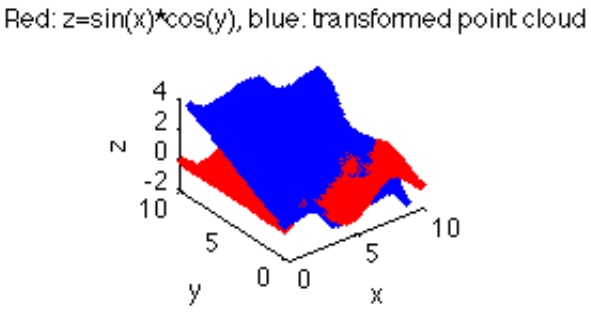


Iterative Closest Point

- Minimize Root Mean Squared Distance between Point Clouds

Source : Mathworks – File Exchange : Iterative Closest Point Package

Scan Matching



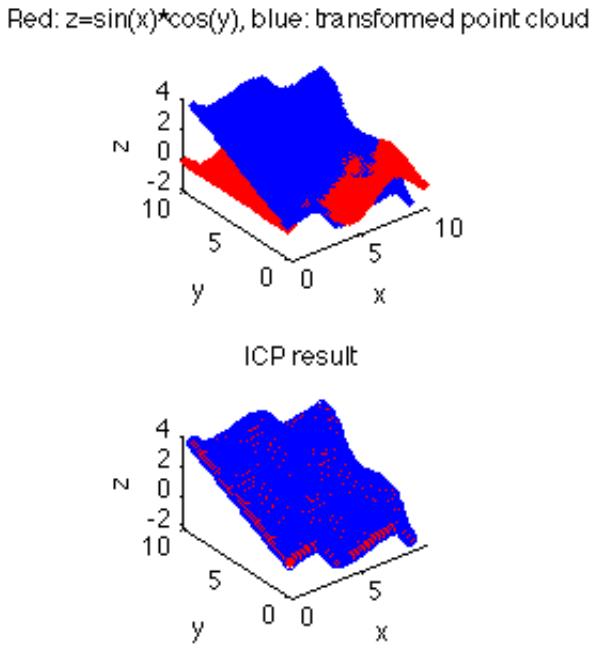
Iterative Closest Point

- Minimize Root Mean Squared Distance between Point Clouds
- Find R & T matrix for transformed Point Cloud w.r.t original cloud.

$$[R,T] = \underset{R,T}{\operatorname{argmin}} \sqrt[2]{\sum_{i=1}^{n} dist(R * Red_{\mathrm{i}} + T, Bluei)^2}$$

Source : Mathworks – File Exchange : Iterative Closest Point Package

Scan Matching



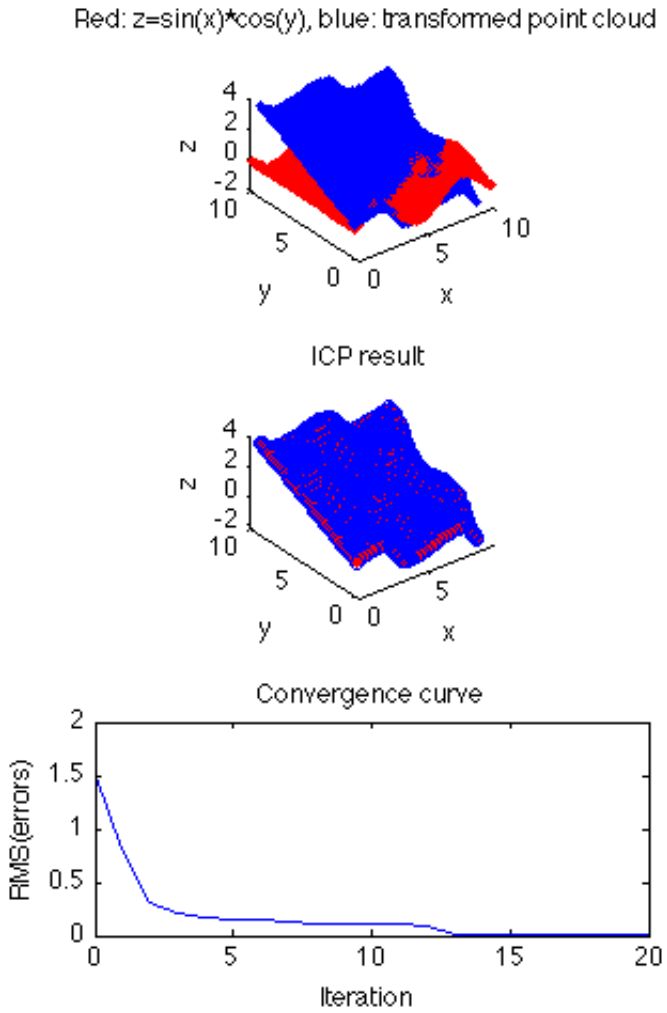
Iterative Closest Point

- Minimize Root Mean Squared Distance between Point Clouds
- Find R & T matrix for transformed Point Cloud w.r.t original cloud.

$$[R,T] = \underset{R,T}{\operatorname{argmin}} \sqrt[2]{\sum_{i=1}^{n} dist(R * Red_{\mathrm{i}} + T, Bluei)^2}$$

Source : Mathworks – File Exchange : Iterative Closest Point Package

Scan Matching



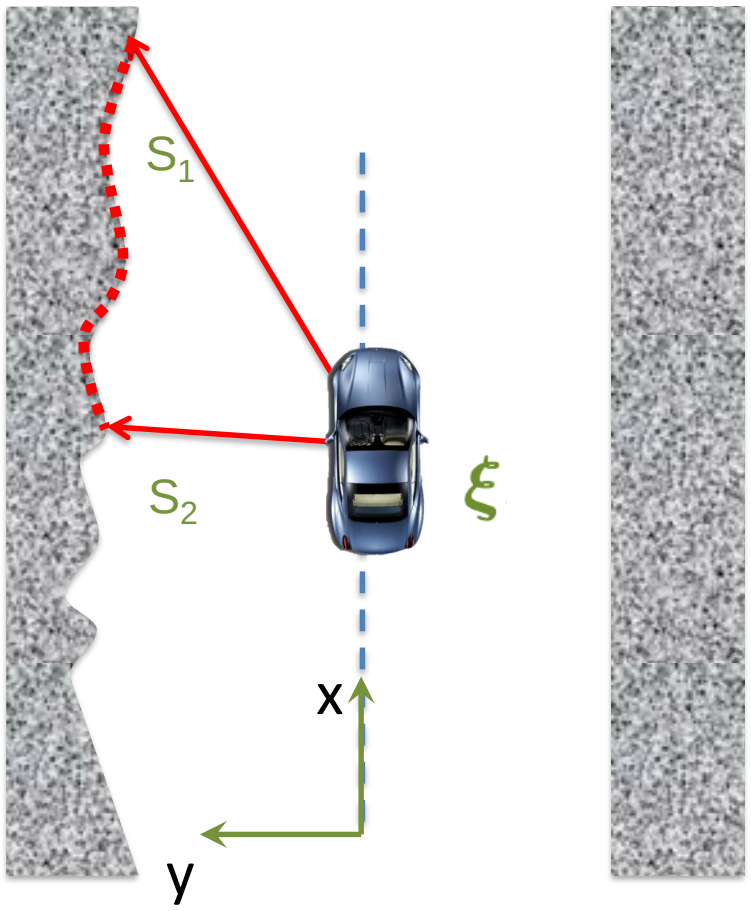
Iterative Closest Point

- Minimize Root Mean Squared Distance between Point Clouds
- Find R & T matrix for transformed Point Cloud w.r.t original cloud.

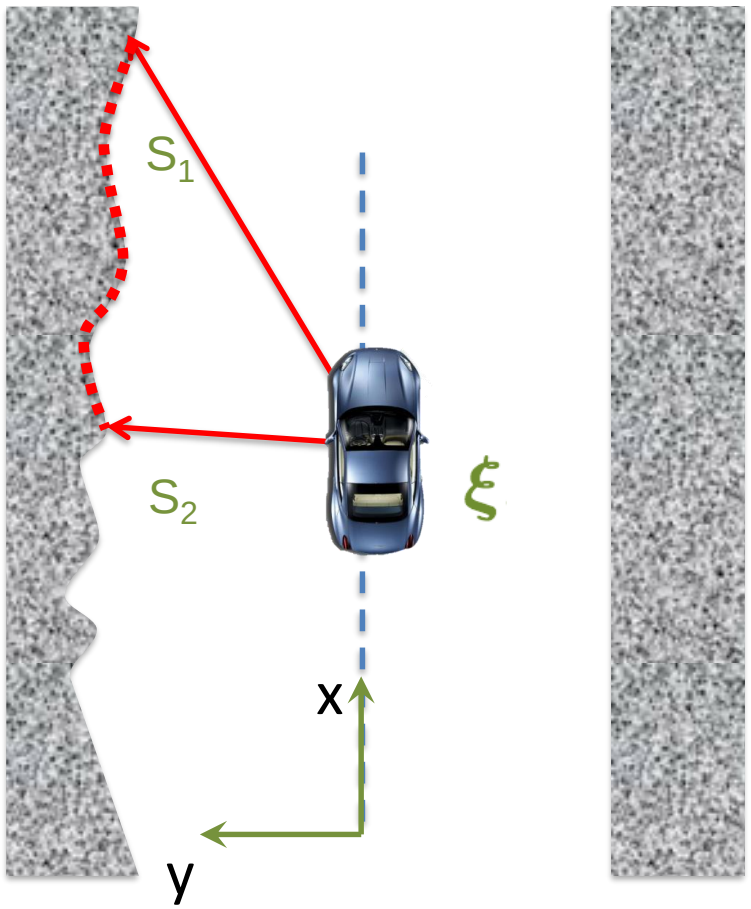
$$[R,T] = \underset{R,T}{\operatorname{argmin}} \sqrt[2]{\sum_{i=1}^{n} dist(R * Red_{\mathrm{i}} + T, Bluei)^2}$$

Source : Mathworks – File Exchange : Iterative Closest Point Package

Scan matching: Hector Slam

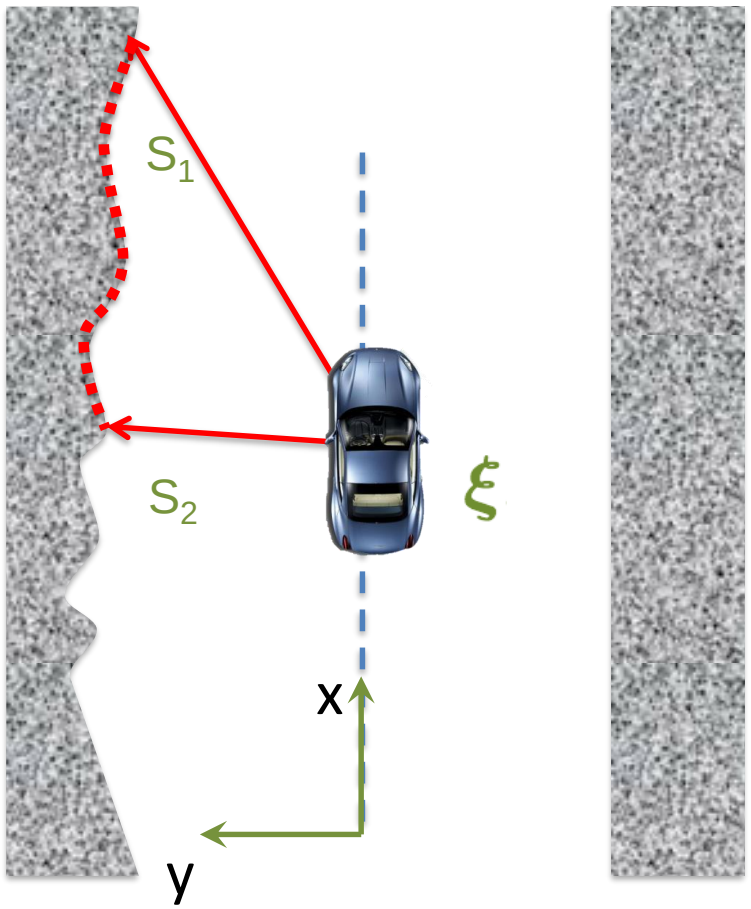


Scan matching: Hector Slam



Robot Pose $\boldsymbol{\xi} = (p_x, p_y, \psi)^{\mathrm{T}}$

Scan matching: Hector Slam



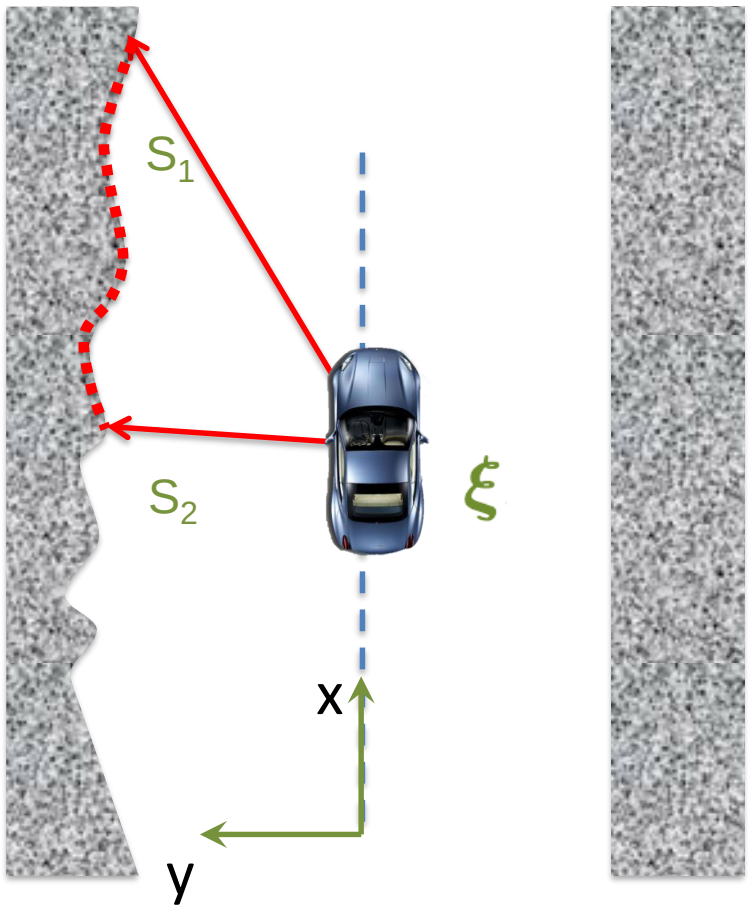
Robot Pose $\boldsymbol{\xi} = (p_x, p_y, \psi)^{\mathrm{T}}$

Impact coordinates of ith scan in world frame

Total of n scans

$$\boldsymbol{\xi}^* = \underset{\boldsymbol{\xi}}{\operatorname{argmin}} \sum_{i=1}^{n} \left[1 - M(\mathbf{S}_i(\boldsymbol{\xi}))\right]^2$$

Scan matching: Hector Slam



Robot Pose $\boldsymbol{\xi} = (p_x, p_y, \psi)^{\mathrm{T}}$

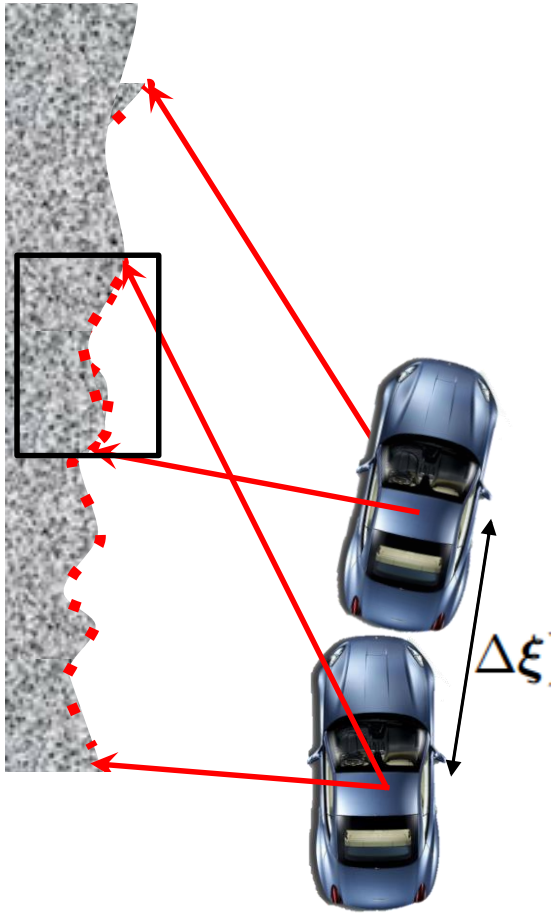
Impact coordinates of ith scan in world frame

Total of n scans

$$\boldsymbol{\xi}^* = \underset{\boldsymbol{\xi}}{\operatorname{argmin}} \sum_{i=1}^{n} \left[1 - M(\mathbf{S}_i(\boldsymbol{\xi}))\right]^2$$

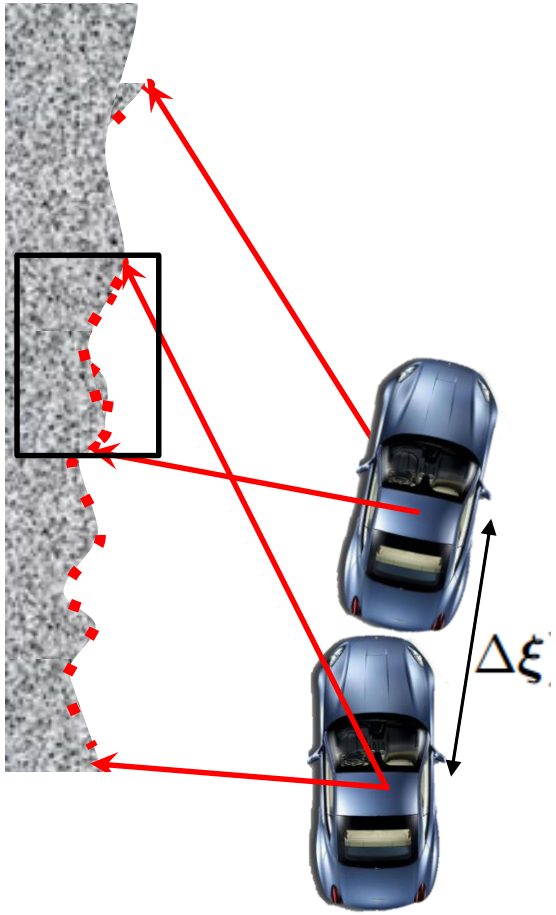
Map Value at coordinates given by S_i

Scan matching: Hector Slam



$$\sum_{i=1}^{n} [1 - M(\mathbf{S}_i(\boldsymbol{\xi} + \Delta\boldsymbol{\xi}))]^2 \to 0\,.$$

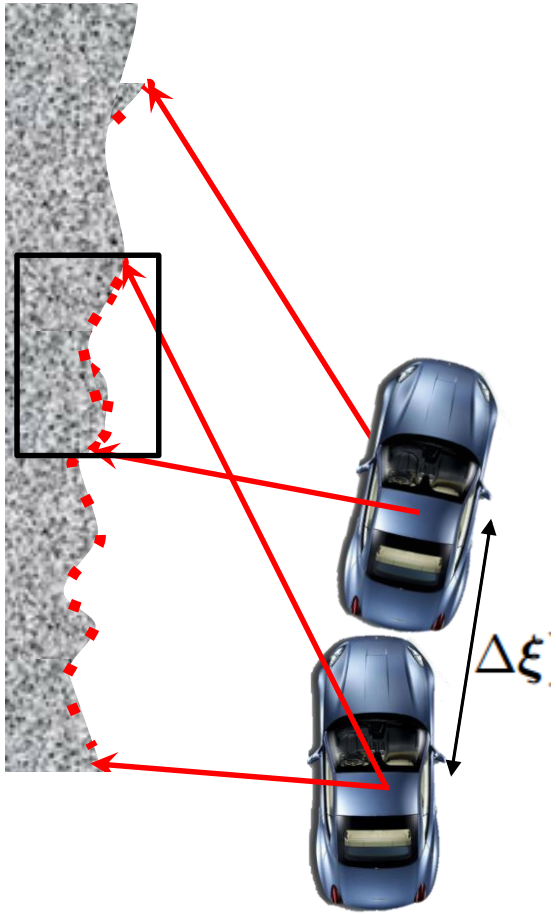
Scan matching: Hector Slam



$$\sum_{i=1}^{n} [1 - M(\mathbf{S}_i(\boldsymbol{\xi} + \Delta\boldsymbol{\xi}))]^2 \to 0\,.$$

Taylor Expansion of Function M

Scan matching: Hector Slam

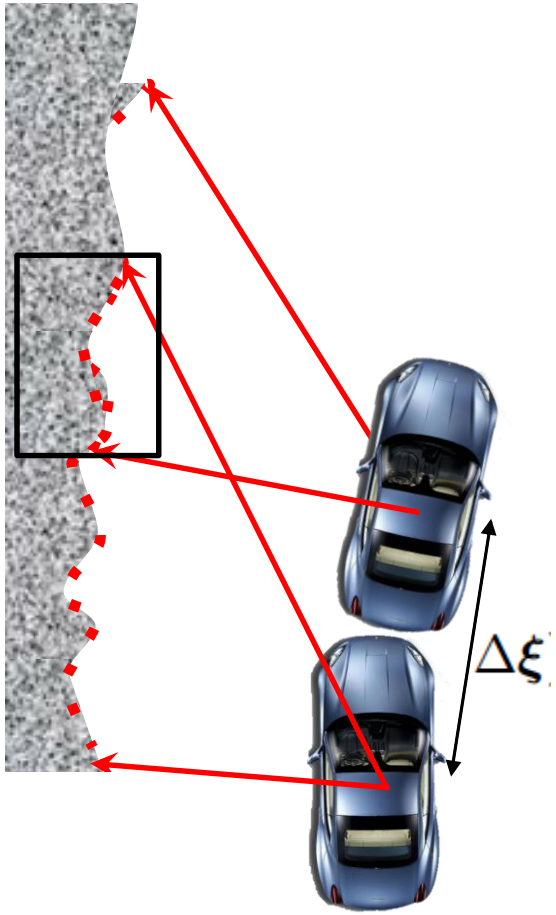


$$\sum_{i=1}^{n} [1 - M(\mathbf{S}_i(\boldsymbol{\xi} + \Delta\boldsymbol{\xi}))]^2 \to 0\,.$$

Taylor Expansion of Function M

$$\sum_{i=1}^{n} \left[1 - M(\mathbf{S}_i(\boldsymbol{\xi})) - \nabla M(\mathbf{S}_i(\boldsymbol{\xi}))\frac{\partial \mathbf{S}_i(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}}\Delta\boldsymbol{\xi}))\right]^2 \to 0$$

Scan matching: Hector Slam



$$\sum_{i=1}^{n} [1 - M(\mathbf{S}_i(\boldsymbol{\xi} + \Delta\boldsymbol{\xi}))]^2 \to 0\,.$$

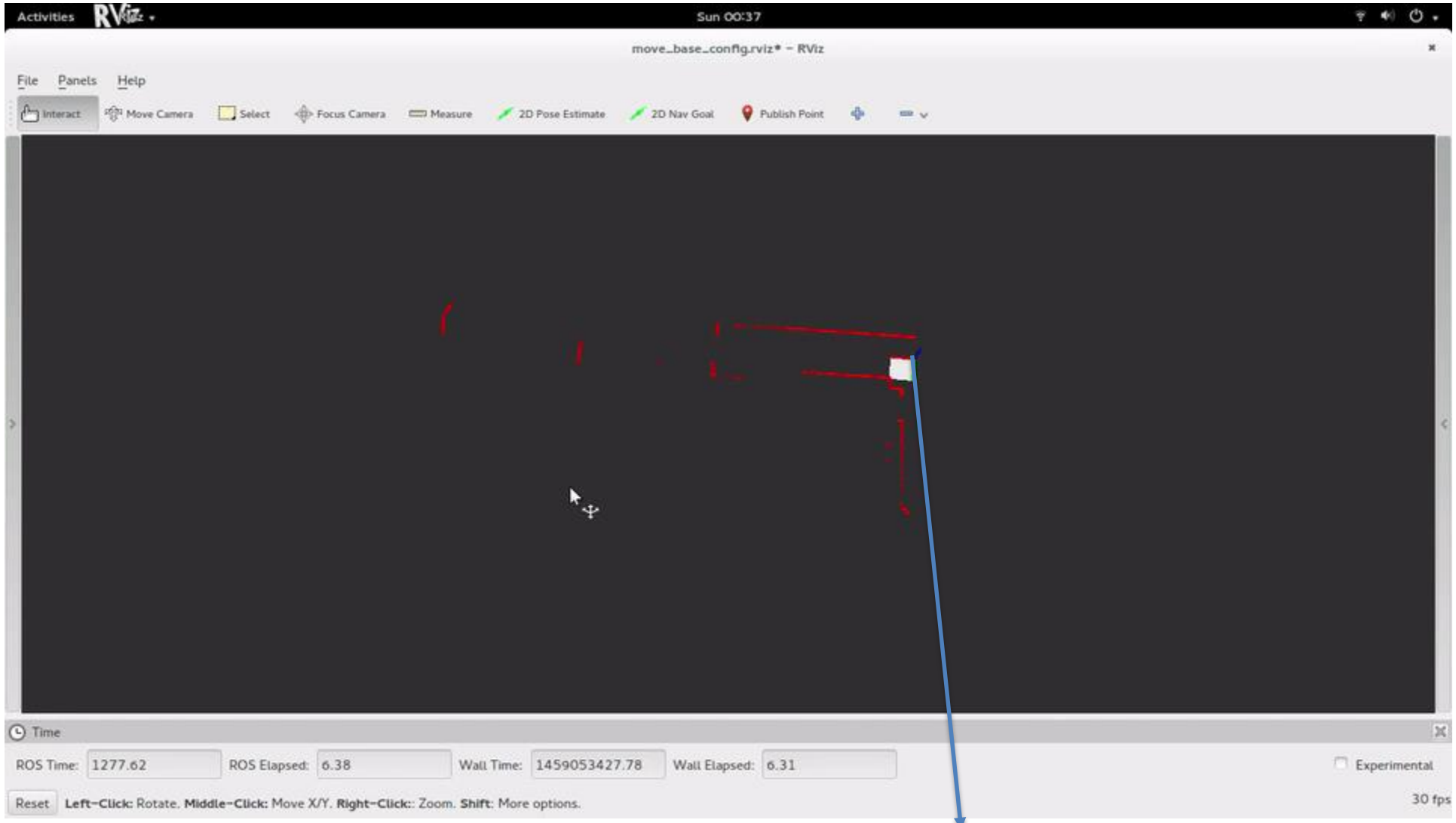
Taylor Expansion of Function M

$$\sum_{i=1}^{n} \left[1 - M(\mathbf{S}_i(\boldsymbol{\xi})) - \nabla M(\mathbf{S}_i(\boldsymbol{\xi}))\frac{\partial \mathbf{S}_i(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}}\Delta\boldsymbol{\xi}))\right]^2 \to 0$$

Solving for $\Delta\xi$ yields Gauss-Newton Equation

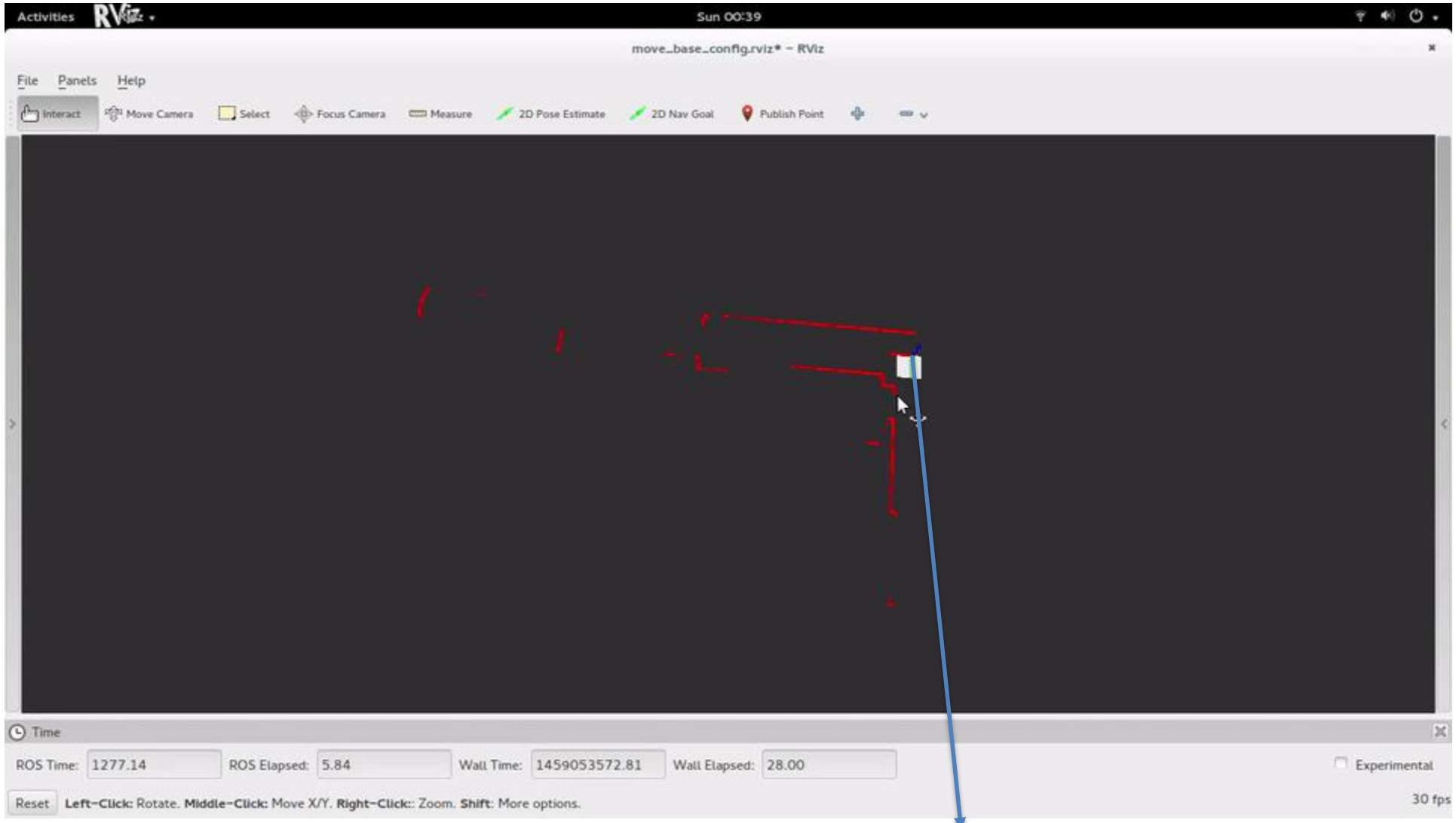
Evaluation of Gauss-Newton equation gives a step $\Delta\xi$ that minimizes the objective function

Raw LiDAR Scans



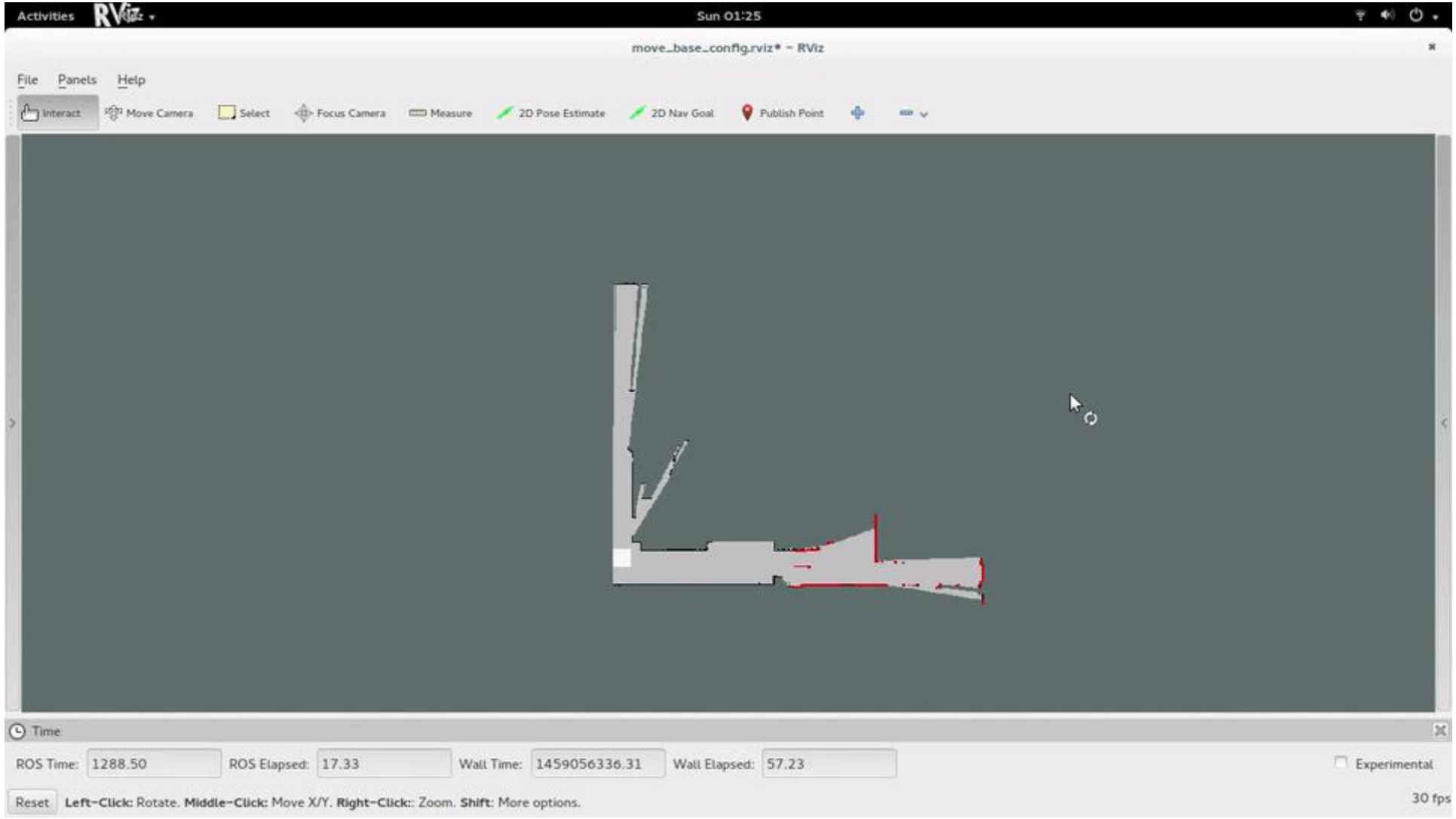
Baseframe Axes

Scans after transforming by $\Delta\xi$ at each stage

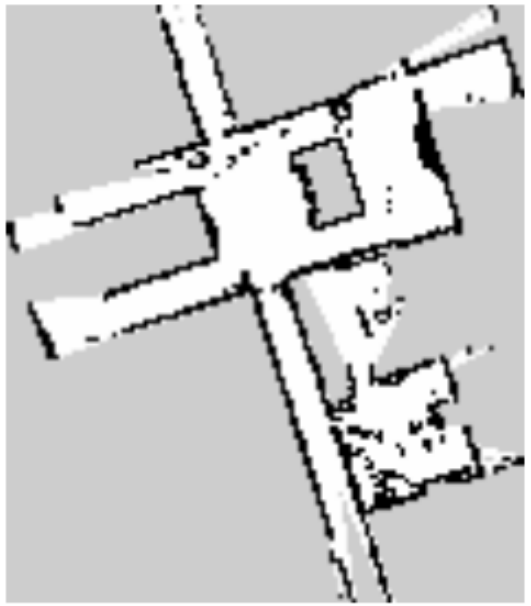


Mapframe Axes

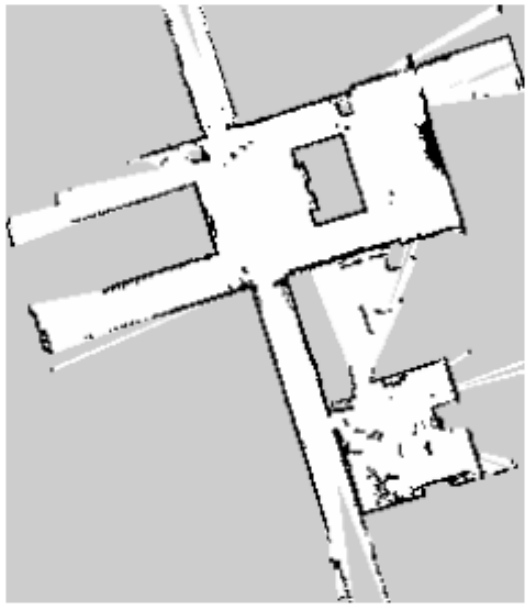
Map Update


Activities
Sun 01:25
move_base_config.rviz* - RViz
File
Panels
Help
Interact
Move Camera
Select
Focus Camera
Measure
2D Pose Estimate
2D Nav Goal
Publish Point
Time
ROS Time: 1288.50
ROS Elapsed: 17.33
Wall Time: 1459056336.31
Wall Elapsed: 57.23
Experimental
Reset
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click:: Zoom. Shift: More options.
30 fps

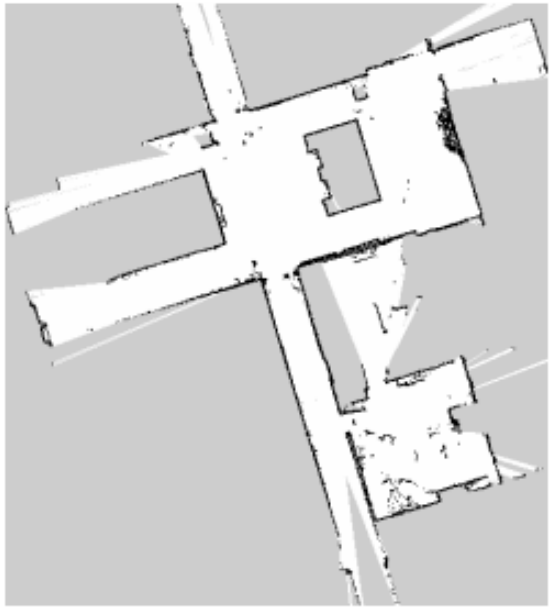
Multi-Resolution Map Representation



20 cm Grid Cell

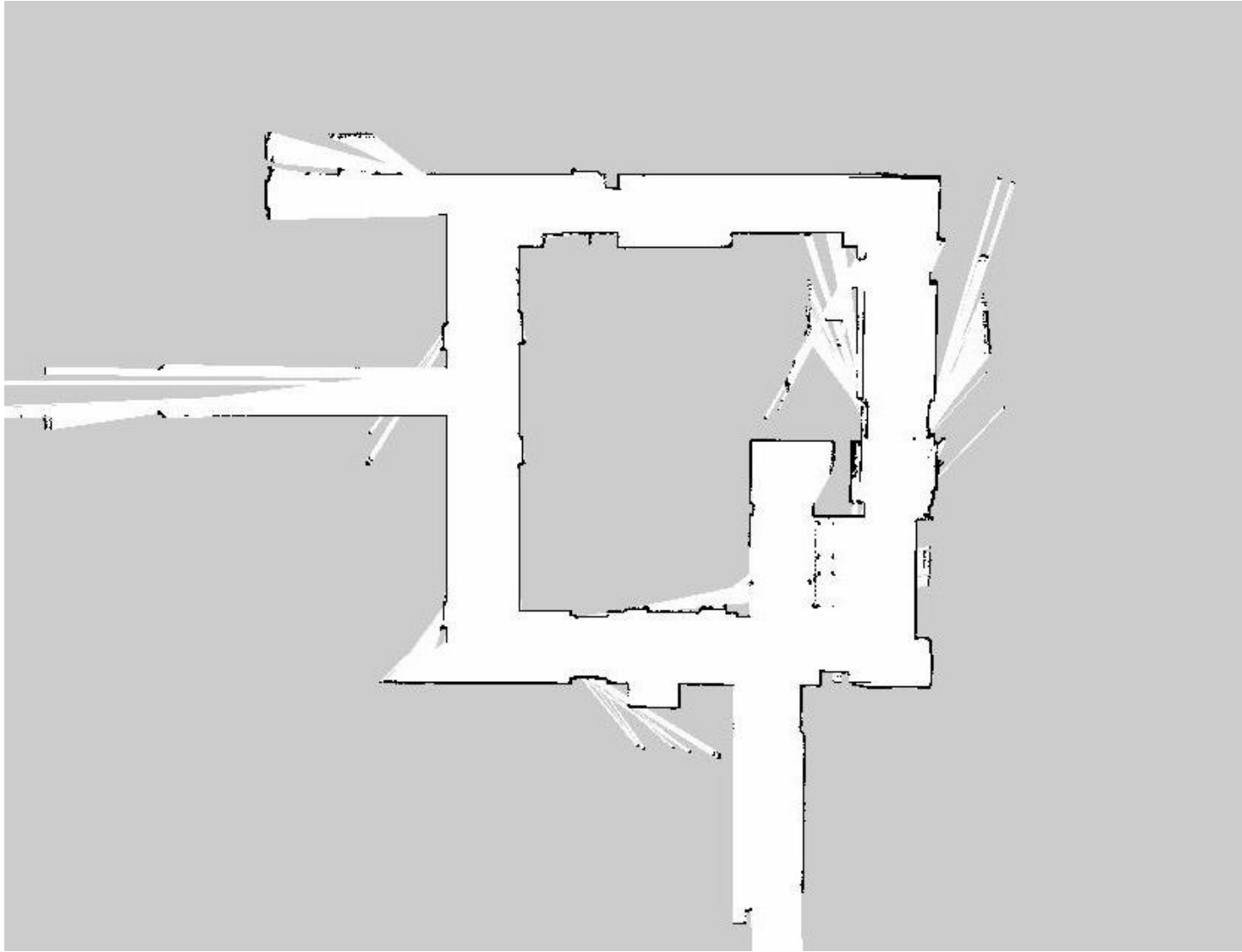


10 cm Grid Cell

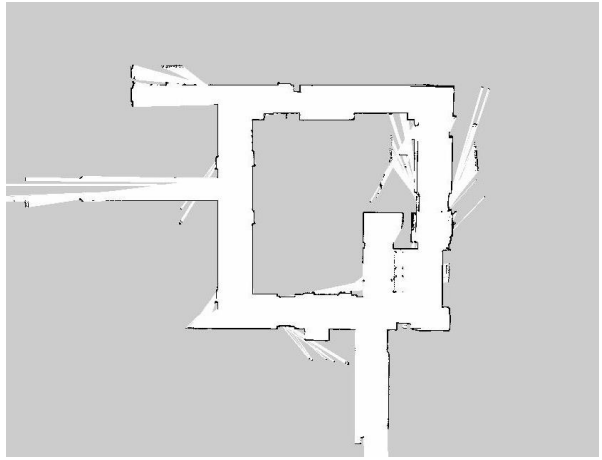


5 cm Grid Cell

Saving the map



Saving the map



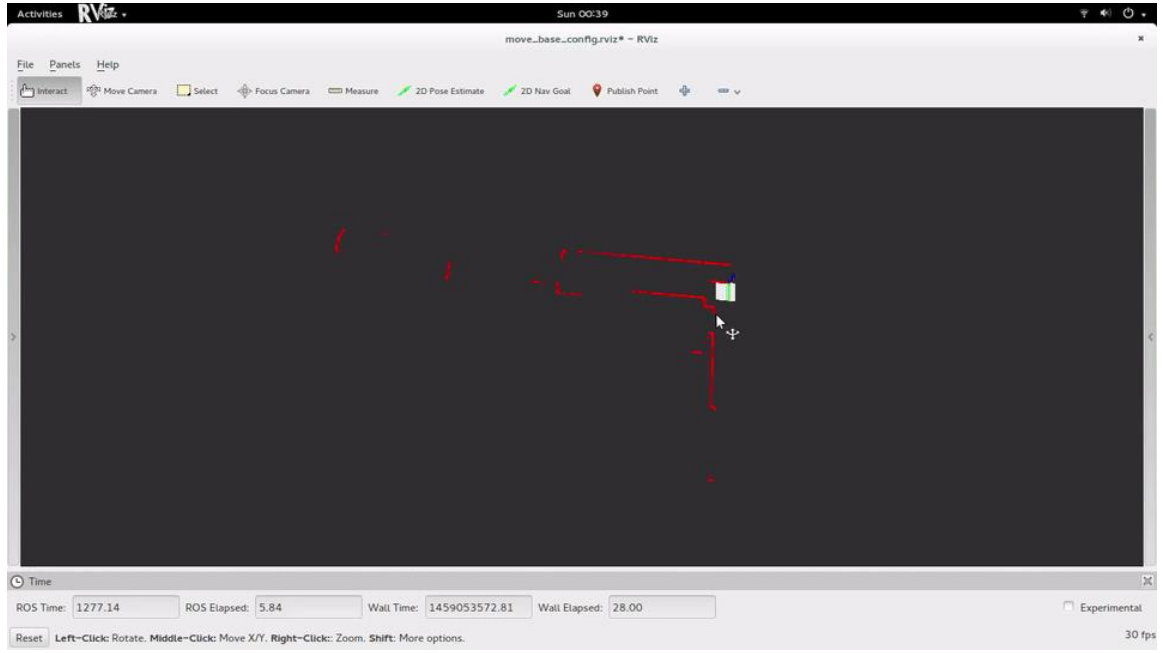
- ROS Package called
  MAP Server

- Allows saving a map currently being published over /map topic

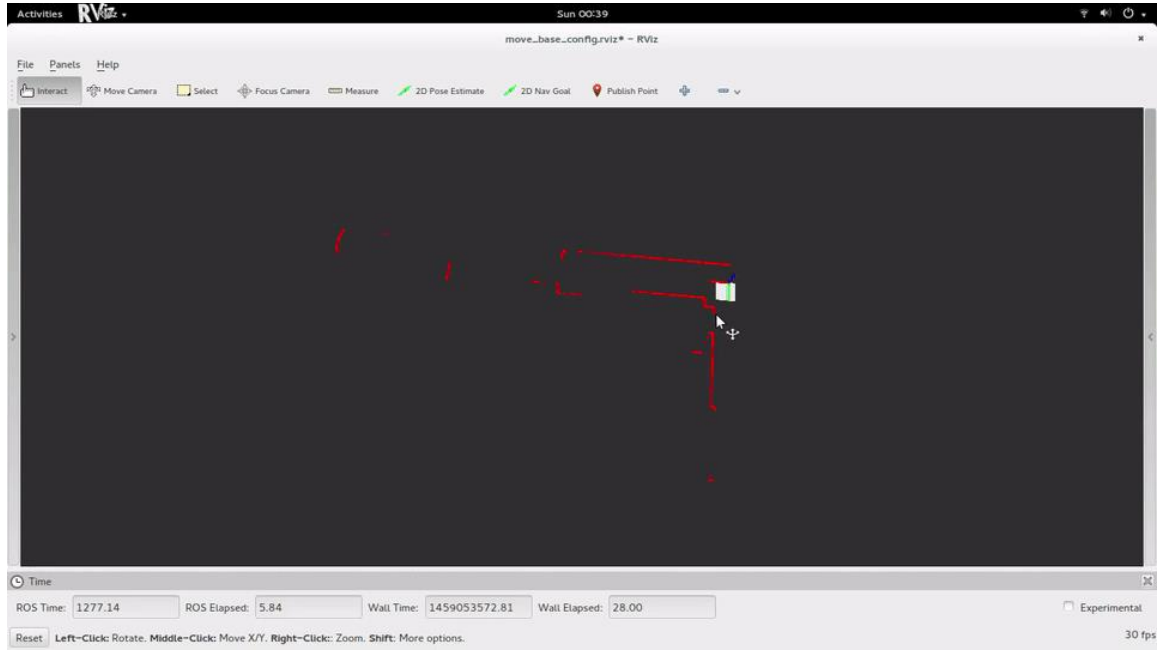
- Save the map:
  rosrun map_server map_saver [-f mapname]

- Load the map:
  rosrun map_server map_server <name.yaml>

Odometry Using Hector Mapping

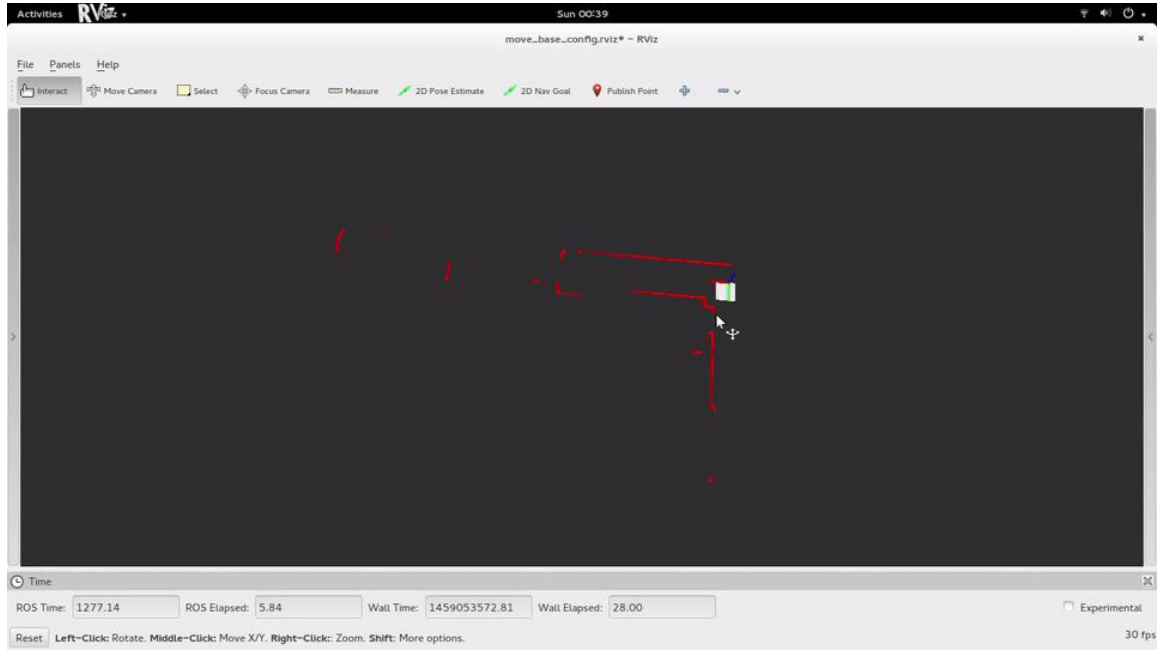


Odometry Using Hector Mapping



- Using Hector Slam for measuring $\Delta\xi$, while discarding the map

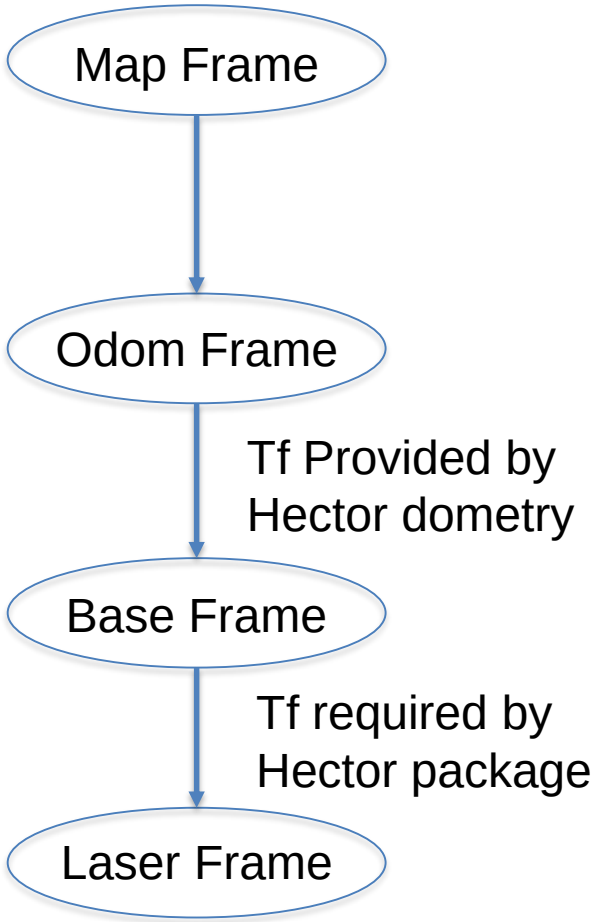
Odometry Using Hector Mapping



- Using Hector Slam for measuring $\Delta\xi$, while discarding the map

- Optional Approach : CSM (Canonical Scan Matcher) by Andrea Censi
  - Scan matching between 2 scans

System Tf tree



Parameters for Hector SLAM : ROS

- map resolution - Grid resolution
- map_update_distance_thresh - minimum distance to be travelled before having a map update
- map_update_angle_thresh - minimum angle to be travelled before a map update
- laser_max_dist - Laser sensor specification
- update_factor_free - Log odds probability for occupied cells
- update_factor_occupied - Log odds probability for free cells

Next Lecture

- Using the map generated today
- Localizing using Adaptive Monte Carlo localization (AMCL)
- Integrating hector odometry and AMCL