MRC 2020
Group 4

FINAL DESIGN PRESENTATION

PICO
in the Hospital

D. van Boven	0780958	B. Kool	1387391
M. Katzmann	1396846	R.O.B. Stiemsma	0852884
R. Konings	1394819	A.S.H. Vinjarapu	1502859

Tutor: Marzieh Dolatabadi Farahani

// Control Systems Group

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Outline

Architecture

- Strategy
- Program structure

Implementation

- Mapping
  - Feature Recognition
  - SLAM (FastSLAM2)
  - Map Updating
- Navigation
  - A* Pathfinding
  - Motion Planning

Architecture : Strategy

Modular division into:

- PICO IO ('interactions')
- World sense ('mapping')
- Planning ('navigation')
- Task Management (main*)

The idea is to minimize backward dependencies.

*: Placement not final

Architecture : Structure

- PICO IO ('interactions')
  - Dependent on the API
  - Lightweight, only remembers sensor data of previous iteration

- World sense ('mapping')
  - Dependent on interactions
  - Maintains an internal world model (lines and points)
  - Recognizes features in the world ('**landmarks**')
  - Performs **SLAM**
  - **Error prone** due to containing a matrix inversion.

Architecture : Structure II

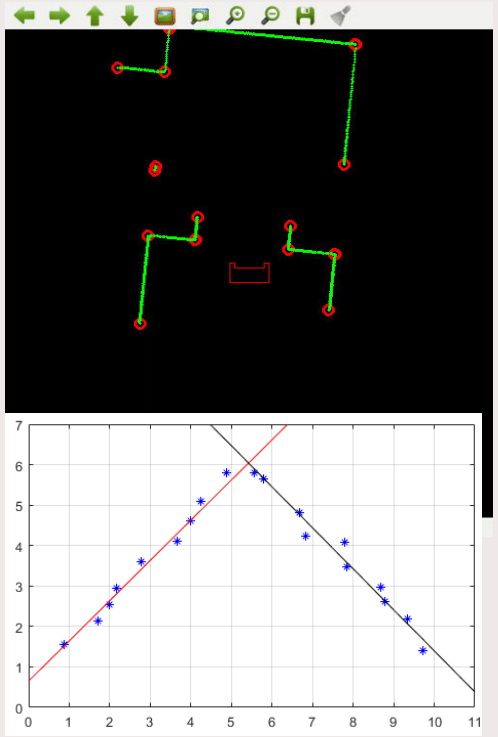
- Planning (‘navigation’)
  - Dependent on mapping

- Task management (main*)
  - Dependent on all the above.
  - Lightly codependent (injects variables and 'modes')
  - Takes intuitive decisions
    ('initialize','scout', 'go', 'wait', 'use cabinet')

*: Placement not final

Implementation: Mapping I – Feature Recognition

- Segmenting laser rangefinder data points
- Total Least Squares regression on a segment's points
- Intersect regression lines for more accurate corner locations
- Match corners to map's landmarks using PICO localization data



*: Placement not final

Implementation: Mapping II – SLAM

- Chosen algorithm is FastSLAM2
  - Monte Carlo localization method
    (very similar to the EKF Particle Filter method)
  - Abuses the static nature of landmarks through Rao-Blackwellization
    → **n 2x2** EKF covariance matrices per hypothesis, vs **nxn**
  - Incorporates measurement into prior position hypothesis

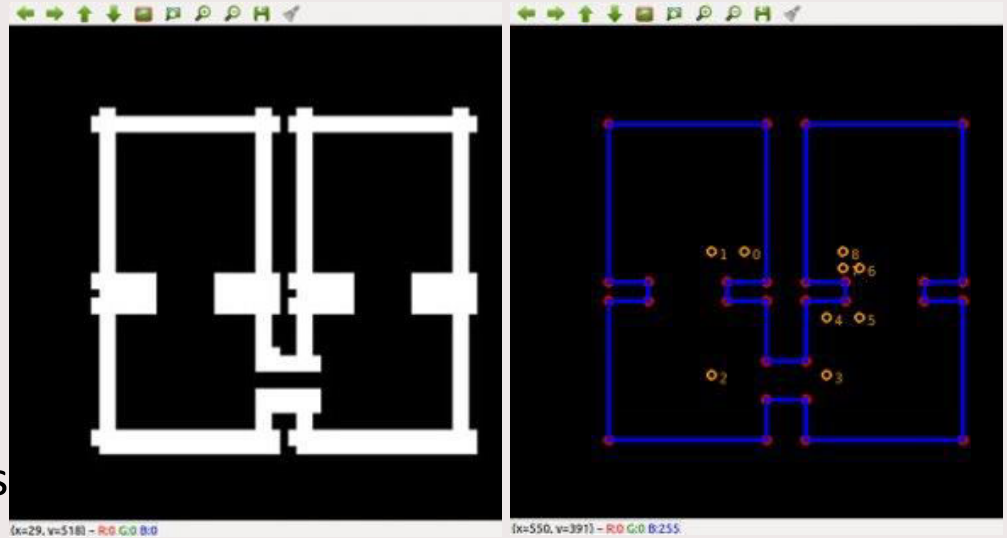
FastSLAM2 promises robust, efficient behaviour for static elements.

Dynamic elements will be accounted for using a confidence-based grid mapping approach.*

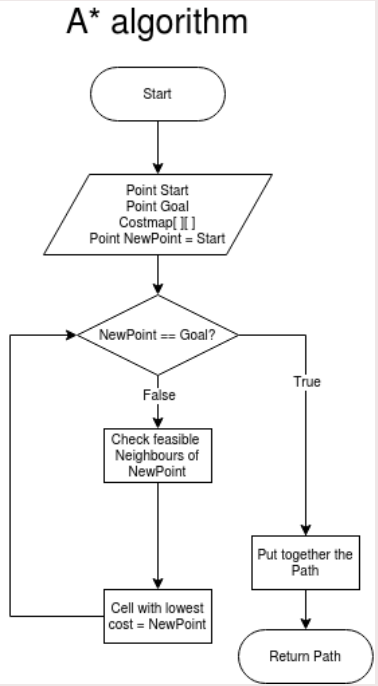
n: the amount of landmarks (corners) in the global map
*: not yet implemented. The idea is a reduced gridSLAM method.

Implementation: Mapping III – Model Updating

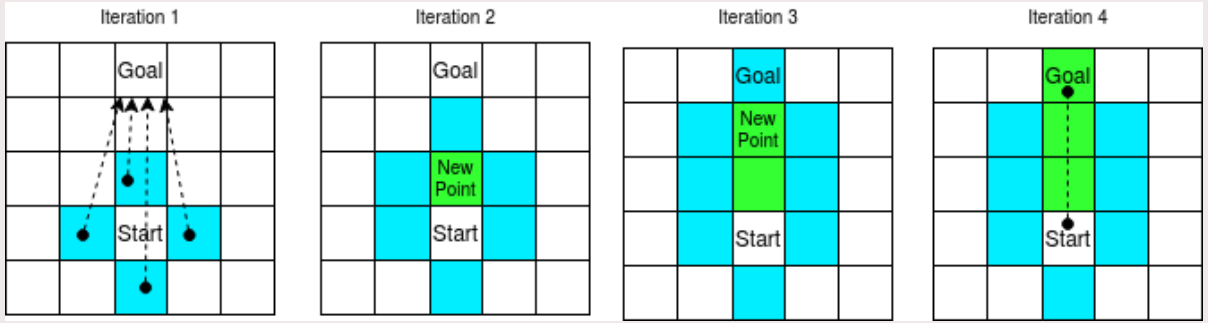
- Model update based on features from FastSlam
- Tile the global map for a costmap
- Uses the aforementioned heatmap for dynamic objects



Implementation: Navigation I – A* Pathfinding



Simple case:



Implementation: Navigation II – Motion Planning

- Trajectory to movement
  - PICO rotates until perpendicular to next point
  - A* trajectory vector translated to x and y speed
- Feedback
  - Current position compared with goal position
  - Position adjusted if necessary