Lecture 6
Robot Navigation

Angelo Cangelosi
Department of Computer Science
The University of Manchester

Content

- Navigation and Manipulation
- Robot Navigation
  - Four questions
- Path planning
- Localisation
- Simultaneous Localisation and Mapping
  - SLAM
  - vSLAM
- Cognitive SLAM: RatSLAM

Robot Motion

- **Navigation**: *Moving the entire robot from one location to another (destination planning)*
  - Locomotion
  - Localisation and mapping

- **Manipulation**: *Moving body part to manipulate the environment*
  - Reaching
  - Grasping

Navigation and Locomotion

- **Mechanical** locomotion
  - Wheels (synchro-drive, omnidirectional wheels)
- **Biomimetic** locomotion
  - Crawling, sliding, running
- **Legged** locomotion
  - Leg events for k legs $N = (2k-1)!$
  - Gait: precomputed coordinated movements (oscillations)

$LeftLeg$	$RightLeg$
U	$-$
$-$	D
$-$	U
D	$-$
U	U
D	D

Robot Navigation: 4 Key Questions

- Where am I going? (and why?)
  (Mission planning)
- What is the best way there?
  Path planning
- Where have I been?
  Map making (Simultaneous Localisation and Mapping)
- Where am I?
  Localisation

Navigation and Path Planning

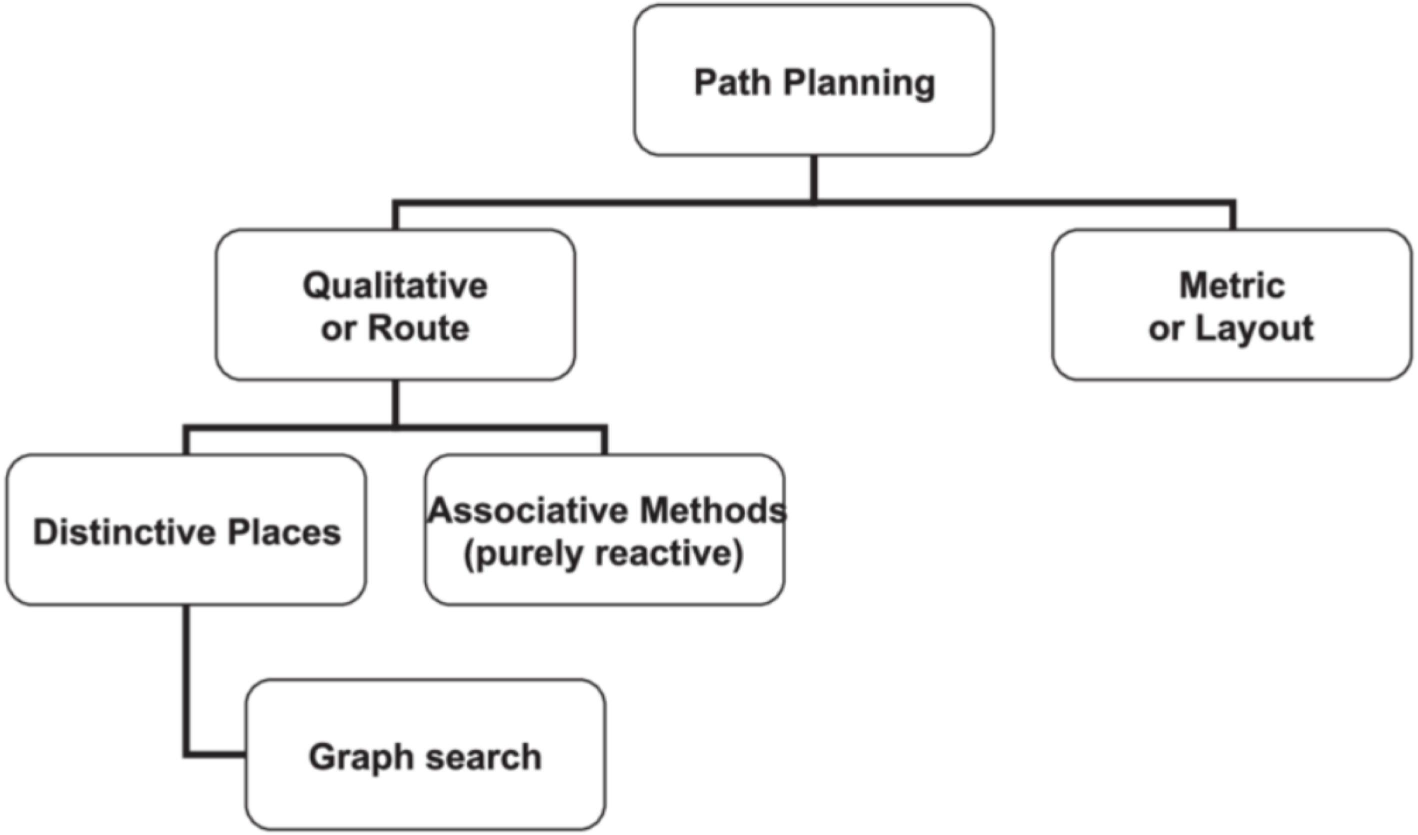
- Four issues of robot navigation
  - Mission planning, **path planning**, mapping and localisation

Path Planning

- **Qualitative** path planning (Topological/Route navigation)
  - Agent’s perspective
  - Topological route with distinctive landmarks
  - Topological route with associations (reactive)
  - AI search and planning
- **Quantitative** path planning (Metric navigation)
  - Metrics or layout map (bird’s eye view map)
  - Orientation- and position-independent

Navigation and Path Planning

- Four issues of robot navigation
  - Mission planning, **path planning**, mapping and localisation



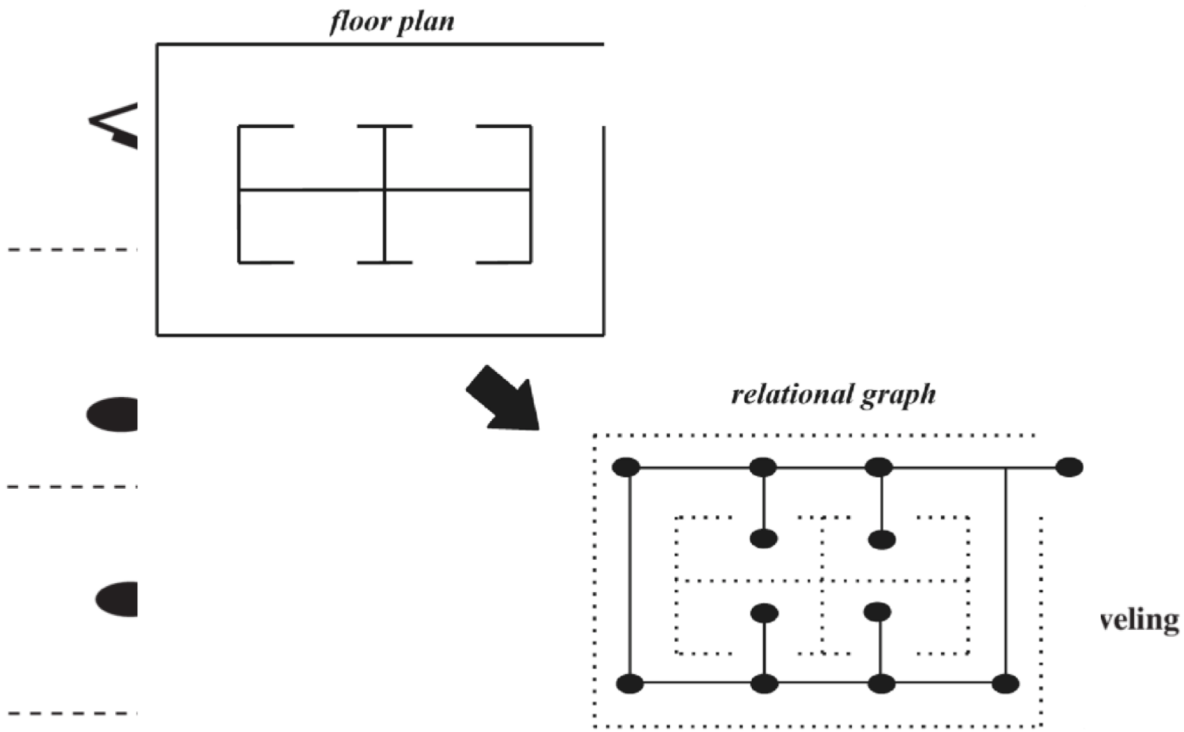
Topological Navigation

- Distinctive places (Landmark methods)
  - Landmarks as waypoints:
    Perceptually distinctive features, easy to recognize
    - Natural landmark: object or scene
    - Artificial landmark: sign or visual pattern (e.g. QR-like)
    - Gateway: landmark to change behavior (e.g. junction)

- Associative methods
  - Association between perceptual state and movement
    - Visual homing: visual patterns associated with locations/routes
      (e.g. bees navigation)
    - QualNav, to infers location from relative positions of landmarks change
      (e.g. constellations)

Topological Navigation: Relational Graph Methods

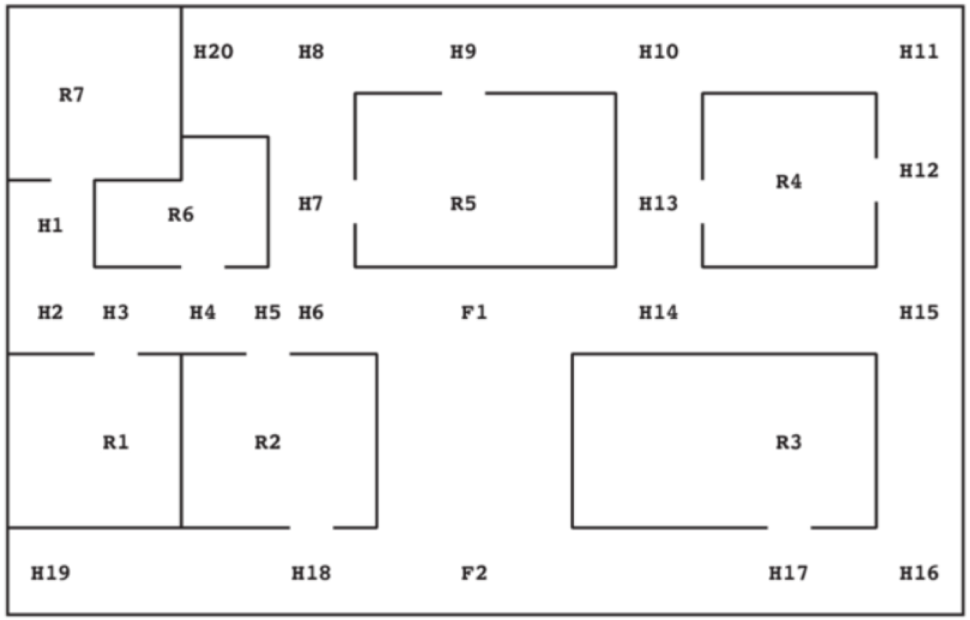
- World as a relational graph of nodes and edges
  - **Nodes** represent gateways, landmarks, or goals.
  - **Edges** represent a navigable path between two nodes
  - Multi-level spatial hierarchy, animal navigation (Bryun & Kuipers)



Topological Navigation

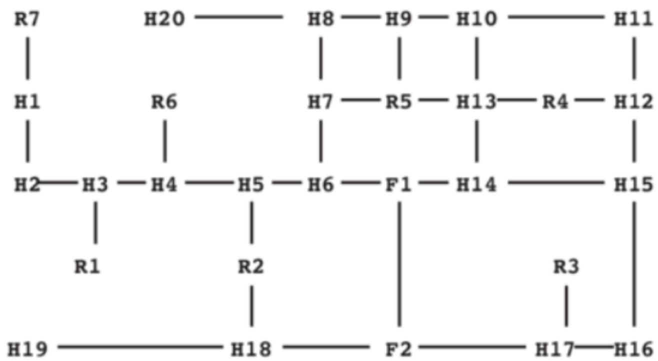
Metric map

3 node types:
- room (R)
- hall (H)
- foyer (F)



a.

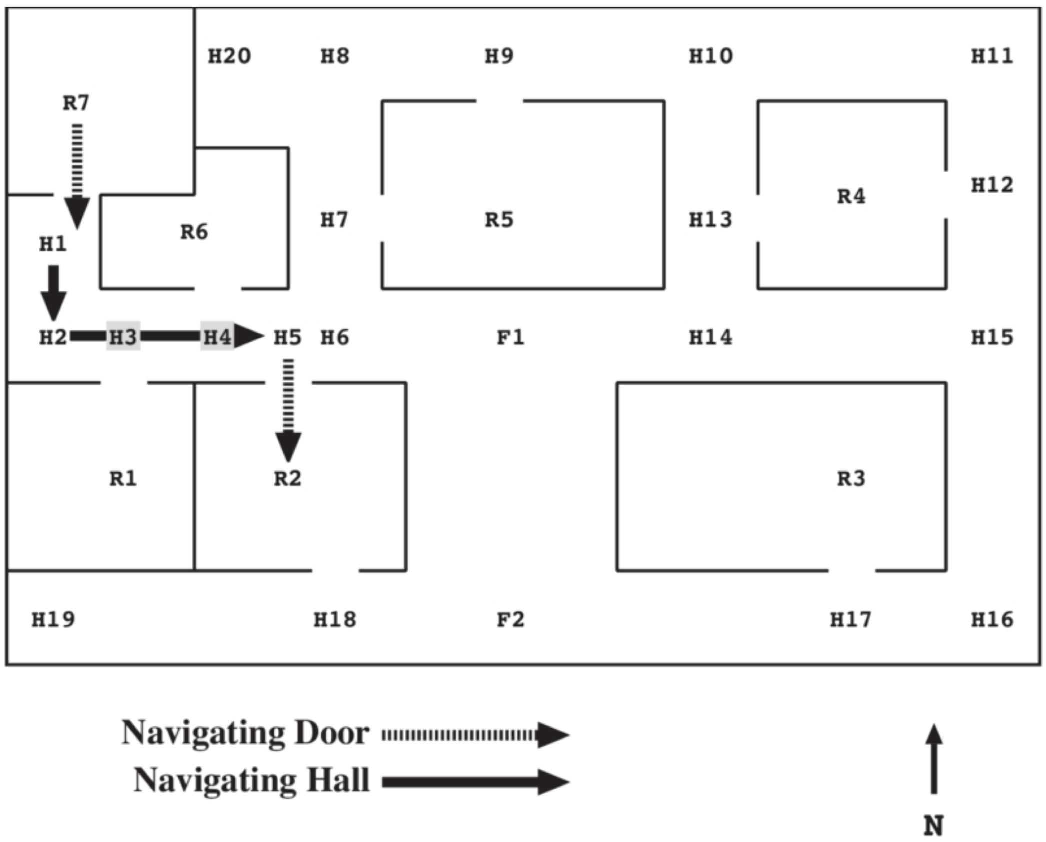
Graphical-topographical representation



Routes from R3 to R7

SFX architecture, 1994 AAAI Mobile Robot Competition (Murphy 2019)

* Topological Navigation Example




```
R7 -> R2
R7 - H1 - H2 - H5 - R2
Moving from R7 to H1, going SOUTH
In navigating door behavior
        ultra looking for door towards the: SOUTH
        MOVE AHEAD MOTOR ACTIVE
        Found door - Initialization terminated
        MOVE THROUGH DOOR MOTOR ACTIVE
Moved through door - Nominal Behavior terminated

Moving from H1 to H2, going SOUTH
In navigating hall behavior
        turning towards the: SOUTH
        Turned towards hall - Initialization terminated
        looking for hall towards the: EAST
        HALL FOLLOW MOTOR ACTIVE
Found hall - Nominal Behavior terminated

Moving from H2 to H5, going EAST
In navigating hall behavior
        turning towards the: EAST
        Turned towards hall - Initialization terminated
        vision looking for door relative: 90 (right side)
        HALL FOLLOW MOTOR ACTIVE
Found door (vision) - Nominal Behavior terminated

Moving from H5 to R2, going SOUTH
In navigating door behavior
        ultra looking for door towards the: SOUTH
        following wall on left (right ground truth)
        WALL FOLLOW MOTOR ACTIVE
        Found door - Initialization terminated
        MOVE THROUGH DOOR MOTOR ACTIVE
Moved through door - Nominal Behavior terminated
Goal location reached!
```

SFX architecture, 1994 AAAI Mobile Robot Competition (Murphy 2019)

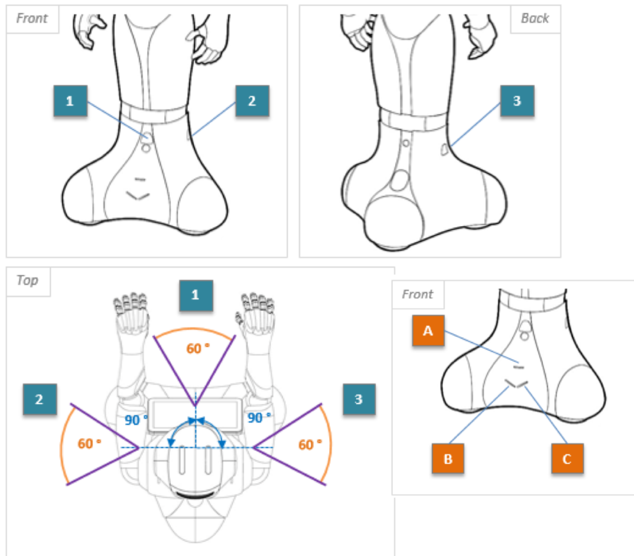
Localisation and Planning

- Challenges
  - How can you *build the map* of a (new) world?
  - How do you *recognise* where you are?
  - How can you *simultaneously* map the world and be sure where you are?
  - How do you *explore new areas* efficiently and consistently?
  - How do you *label* the map with objects and features or terrain?

SLAM: Simultaneous Localisation and Mapping

Complex Robot Sensors: Laser/LIDAR

- Laser range sensors: LIDAR
  LIght **D**etection and **R**anging
  - (aka Laser radar, Ladar).
  - For indoor navigation and mapping
  - Emit highly amplified and coherent radiation at target frequency(ies)
  - Planar 2D horizontal
  - Time-of-flight principle (fast!)
  - Phase-shift measurements (for short-range distances)



Robot Localisation

- Mobile robot localization (position estimation)
  - "The problem of determining the pose of a robot relative to a given map of the environment." (Thrun et al.)
  - Pose **x** of a robot: its position (x, y) and orientation (θ)

    x$= (x, y, \theta)^T$

Robot Localisation

- Mobile robot localization (position estimation)
  - "The problem of determining the pose of a robot relative to a given map of the environment." (Thrun et al.)
  - Pose **x** of a robot: its position (x, y) and orientation (θ)

    x$= (x, y, \theta)^T$

  - **Bayesian algorithms** to estimate probable location (e.g. Monte Carlo)
    - Feature-based localization (Marcov, Extended Kalman Filters)
    - Iconic localization (Monte Carlo localisation / Particles)

Feature-Based Localisation

- To extract features from raw data and match these to the map.
  - e.g. corners, walls, doors
  - Computational efficient (world abstraction)
- Common methods:
  - Marcov localization
  - Extended Kalman Filters

Marcov Localisation

Typically used for **global** localisation without initial known pose

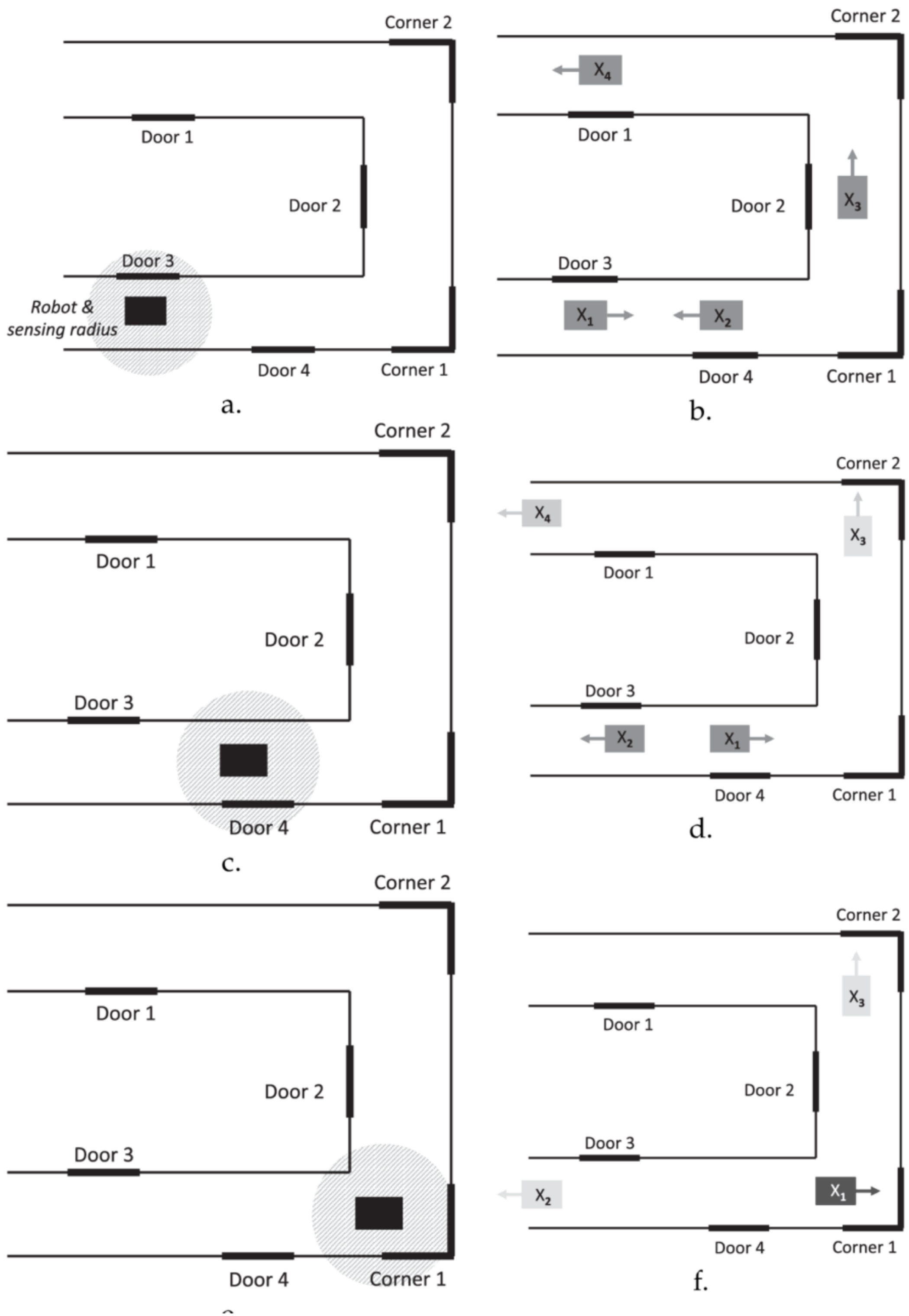
Markov localization computes the belief in each of the possible poses, with the highest belief being the most likely pose. The belief is a function,

$$bel(\mathbf{x}_t) = f(bel(\mathbf{x}_{t-1}), \mathbf{u}_t, \mathbf{z}_t, m)$$

where

- $\mathbf{x}$: is the set of possible poses; in practice the space may be discretized
- $bel(\mathbf{x}_t)$: is the belief that the robot is at $\mathbf{x}$ at time t
- $bel(\mathbf{x}_{t-1})$: is the belief that the robot was at $\mathbf{x}_{t-1}$ in the previous time step
- $\mathbf{u}_t$: is the set of control actions or what movements the robot is commanded to execute at time t
- $\mathbf{z}_t$: is the set of measurements or what the robot observed at time t
- m: is the map.

(Murphy 2019)


Corner 2
Door 1
Door 2
Door 3
Robot &
sensing radius
Door 4
Corner 1
a.
Corner 2
X_4
Door 1
Door 2
X_3
Door 3
X_1
X_2
Door 4
Corner 1
b.
Corner 2
Door 1
Door 2
Door 3
Door 4
Corner 1
c.
Corner 2
X_4
X_3
Door 1
Door 2
Door 3
X_2
X_1
Door 4
Corner 1
d.
Corner 2
Door 1
Door 2
Door 3
Door 4
Corner 1
Corner 2
X_3
Door 1
Door 2
Door 3
X_2
X_1
Door 4
Corner 1
f.

(Murphy 2019)

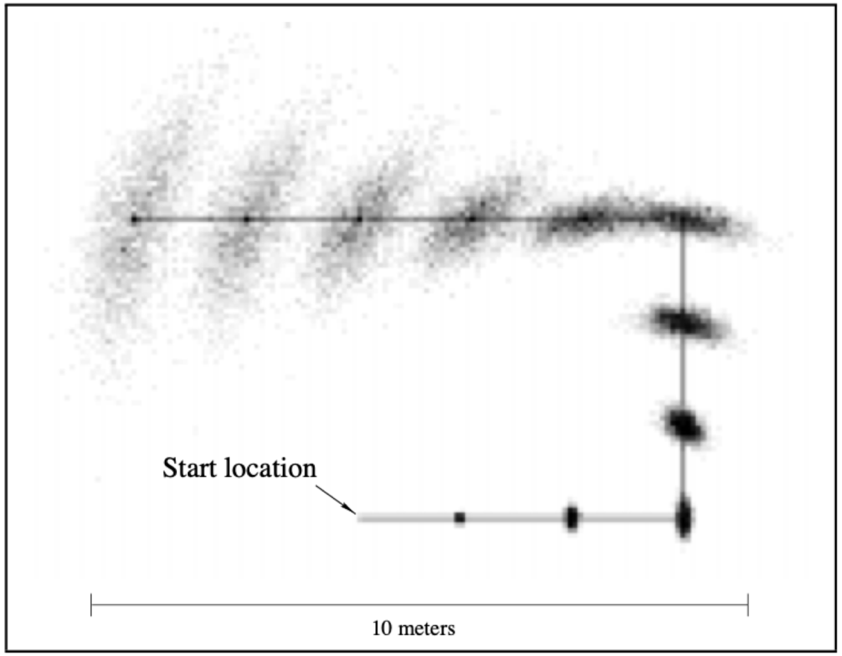
Extended Kalman Filters

- Typically used for **local** localisation
- To **predict** what the robot will sense at the next timestep, given the control action
  - Take difference between the prediction and what it actually senses, to correct/finetune its estimates (e.g. adjust for drift)
  - Extract a set of observations about the features z
  - Match those features to the map (correspondence variables)

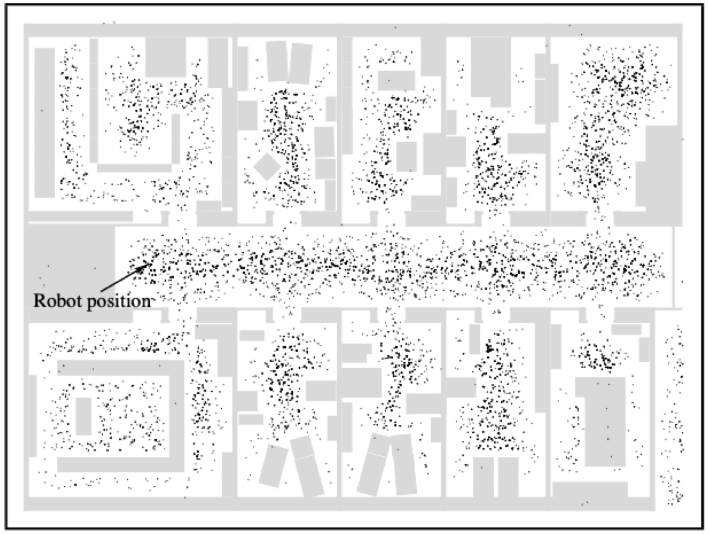
Iconic Localisation

- Raw sensor readings to **match** actual observations to expected **observations**
  - High computation due to many sensor readings
- Grid-based method
  - World partitioned into tesselation of convex polygons
  - Compute the likelihood of all possible poses within each polygon, given the observation
- **Monte Carlo localization**
  - **Particles**, or sample poses, are scattered through space
  - Compute the belief that the robot is at the pose
  - Next, more particles are added, and some particles “die” if they have a low probability

Iconic Localisation: Monte Carlo Particles



Non sensing robot



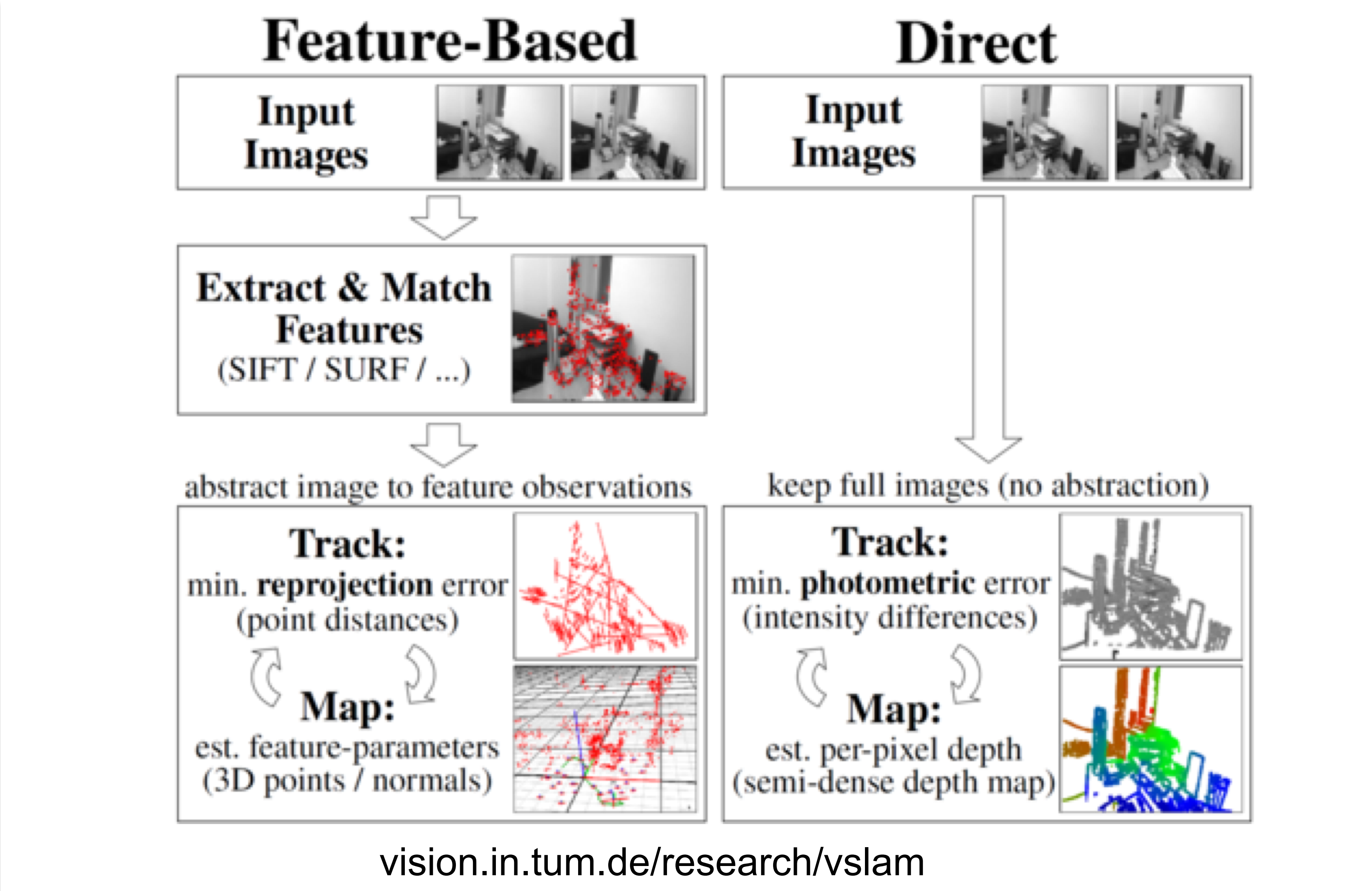
Fox et al. (1999) Monte Carlo Localization: Efficient Position Estimation for Mobile Robots

SLAM: Simultaneous Localisation and Mapping

- Build map whilst localising (new environment)
- Rao-Blackwellized Particle Filter
  - Each particle presents a path and a **local map**
  - At each observation, update only the sensed area of the maps and compute the belief in each particle
  - Use tree to save all the particles that form the history of the current set of particles
- Loop closure
  - To check and adjust map for misalignments
  - path will propagate backwards and improve the overall map

Videos: https://www.youtube.com/watch?v=ykQvQrcE3Xk
https://www.youtube.com/watch?v=qsNGoHi7o2U
https://www.youtube.com/watch?v=OV6wNr62nqQ

Visual SLAM (vSLAM)



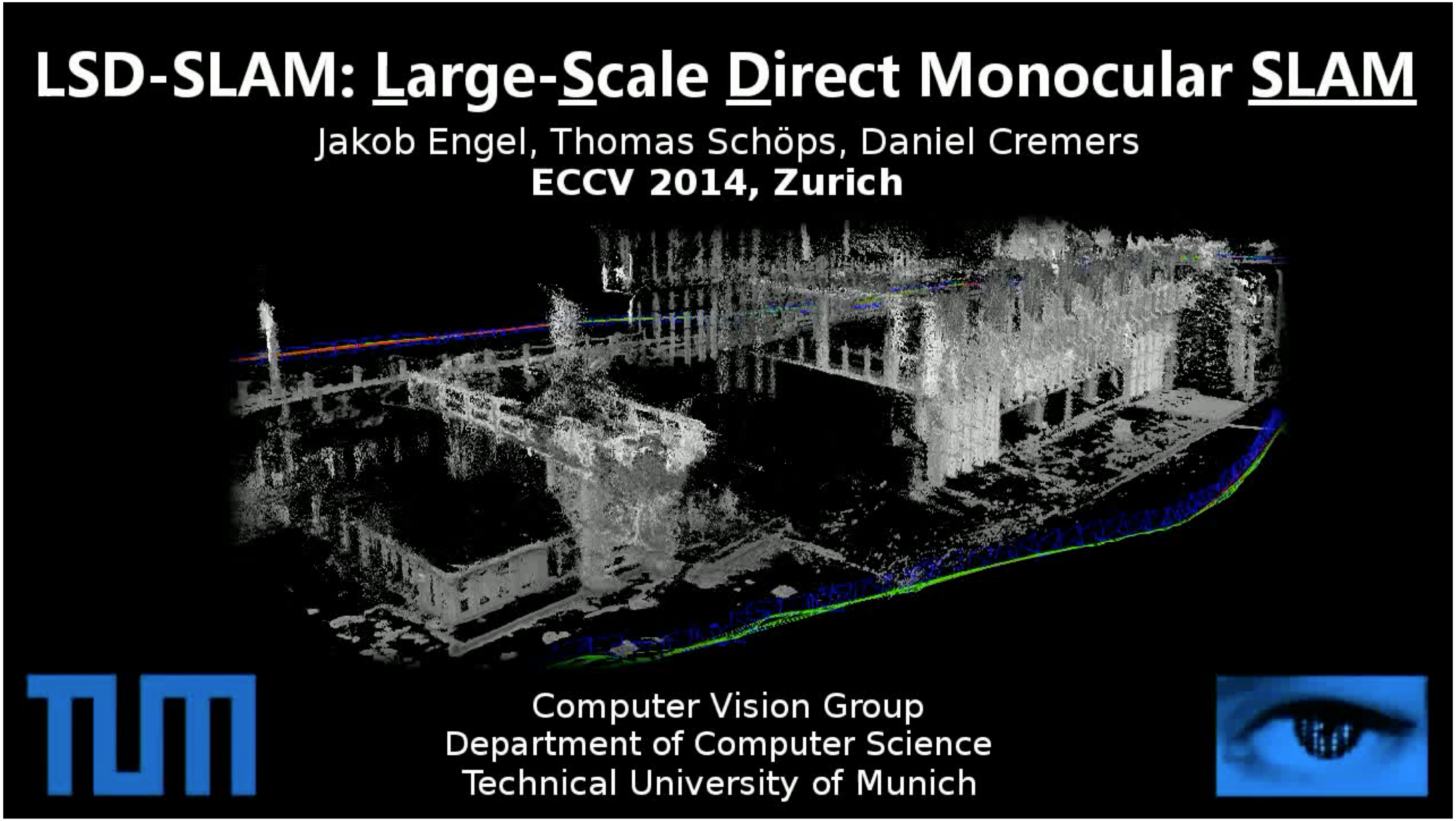
[Taketomi, Uchiyama & Ikeda (2017). Visual SLAM algorithms: a survey from 2010 to 2016. *IPSJ*]

Visual SLAM (vSLAM)

	Method	Map density	Global optimization	Loop closure
Mono-SLAM [26]	Feature	Sparse	No	No
PTAM [15]	Feature	Sparse	Yes	No
ORB-SLAM [38]	Feature	Sparse	Yes	Yes
DTAM [43]	Direct	Dense	No	No
LSD-SLAM [21]	Direct	Semi-dense	Yes	Yes
SVO [51]	Semi-direct	Sparse	No	No
DSO [53]	Direct	Sparse	No	No
KinectFusion [57]	RGB-D	Dense	No	No
Dense visual SLAM [23]	RGB-D	Dense	Yes	Yes
ElasticFusion [66]	RGB-D	Dense	Yes	Yes
SLAM++ [60]	RGB-D	Dense	Yes	Yes

Taketomi, Uchiyama & Ikeda (2017). Visual SLAM algorithms: a survey from 2010 to 2016. *IPSJ*

Visual SLAM (vSLAM)



https://vision.in.tum.de/research/vslam

Cognitive Navigation

RatSLAM

- Emulating the spatial navigation ability of the rat's hippocampal system
- To build and use maps simultaneously of large and complex environments
  - less accurate than SLAM, but flexibility to
    - cope with noisy input
    - deal with changing environments
    - accommodate complexity

Milford & Wyeth (2010). Persistent Navigation and Mapping Using a Biologically Inspired Slam System. IJRR, 29, 9 , 1131–53

Navigation in Rats

- Discoveries of **place cell** (O'Keeffe 1971), **head direction cell** (Ranck 1985), and **grid cell** (Moser 2005) in hippocampus
- Grid cells and place cells form **path integration mechanism** for cognitive map
  - These give rats with innate sense of the world and of their location within it
  - Similar to SLAM mechanisms

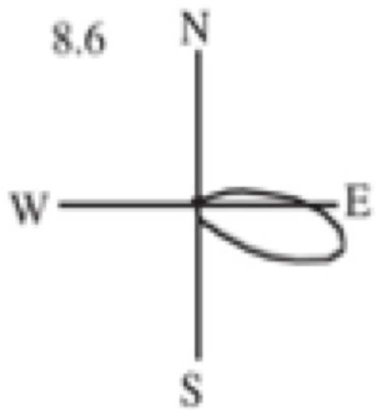
Place Cells

- Place cells fire consistently when the rat is at a **specific location** in the environment
  - for recognition of specific familiar place (‘firing field’)
  - external input about environmental/event
  - internal input from inner path integration system (self-motion)

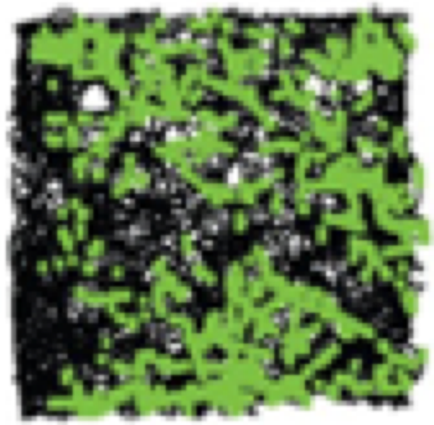
- Cognitive map in the hippocampus
  - thousands of place cells, covering the surface of any space, act as a mapping system

Head Direction Cells

- Head direction cells fire when the rat's head is at **specific orientations**
  - All orientations are represented by the head direction cell population
  - Not correlated to the location of the animal's body



Tuning curve of head direction cell
in polar coordinates

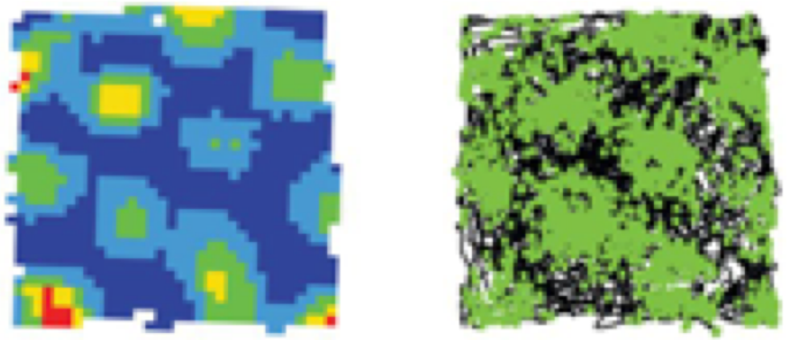


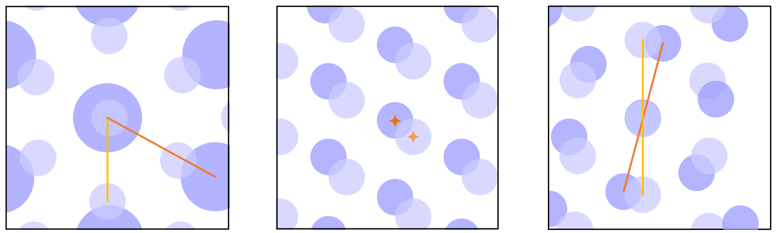
Path traveled (black) and
firing of head direction cell (green)

O'Keefe 2019

Grid Cells

- Grid cells fire in a **metrically regular way** across the whole surface of a given environment
  - Like place cell but with multiple firing fields
  - Cell fires when rat is located at any of the vertices of a **hexagonal pattern** across the environment
  - Cells differ from each other in *spacing*, *phase* and *orientation*





- Cells as signal for measuring displacement distances and direction (metric)

Path Integration

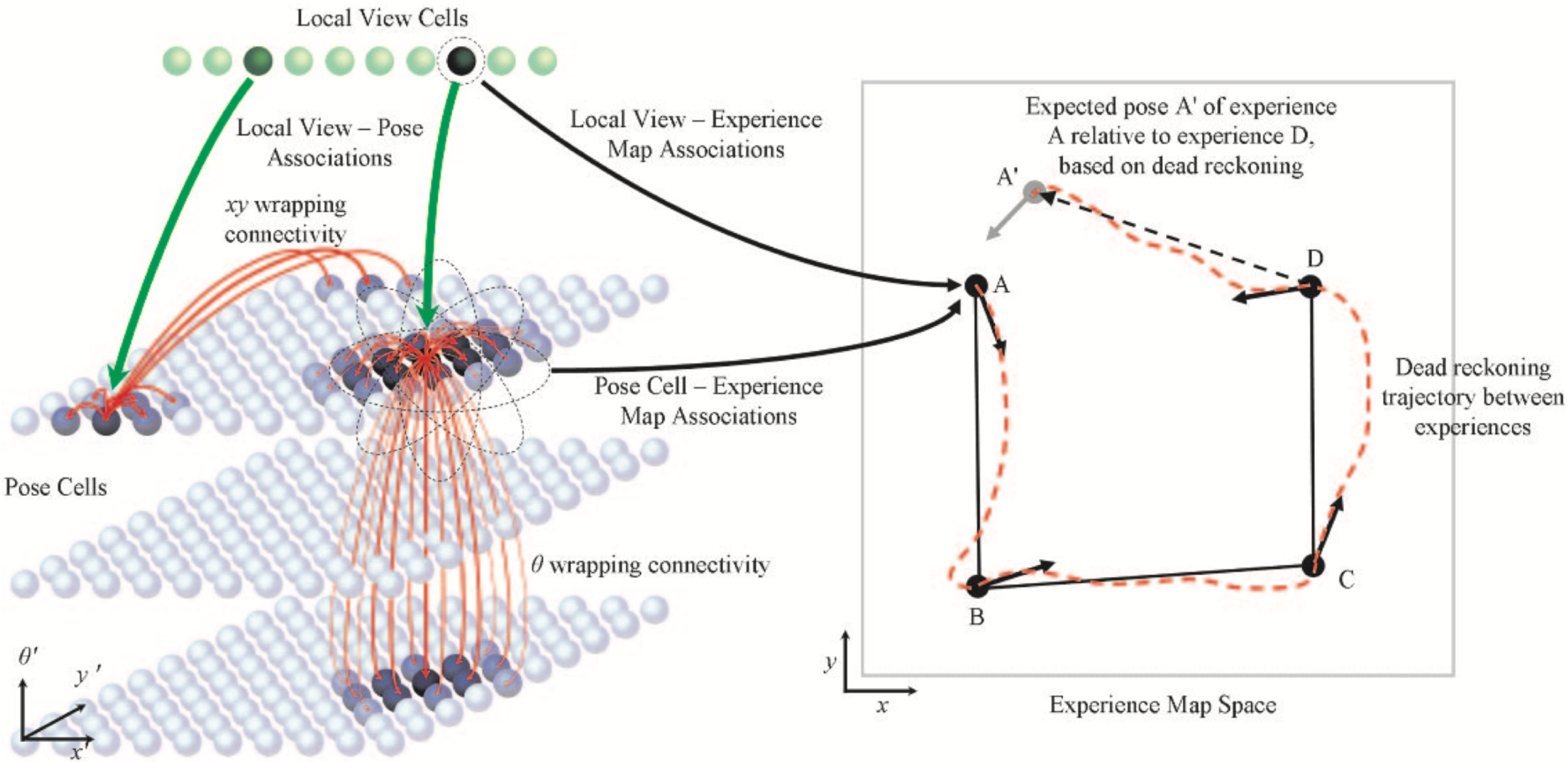
- Inner path integrator for **forming inner cognitive maps** of environments
  - to know which direction it is moving, at what speed, and for how long
- Place cell population memorises sequence of discrete positions: a **cognitive map**
  - Rats can correct accumulative movement errors
  - When a rat returns to a familiar environment, the path integrator resets to adjust to perceived environment

RatSLAM

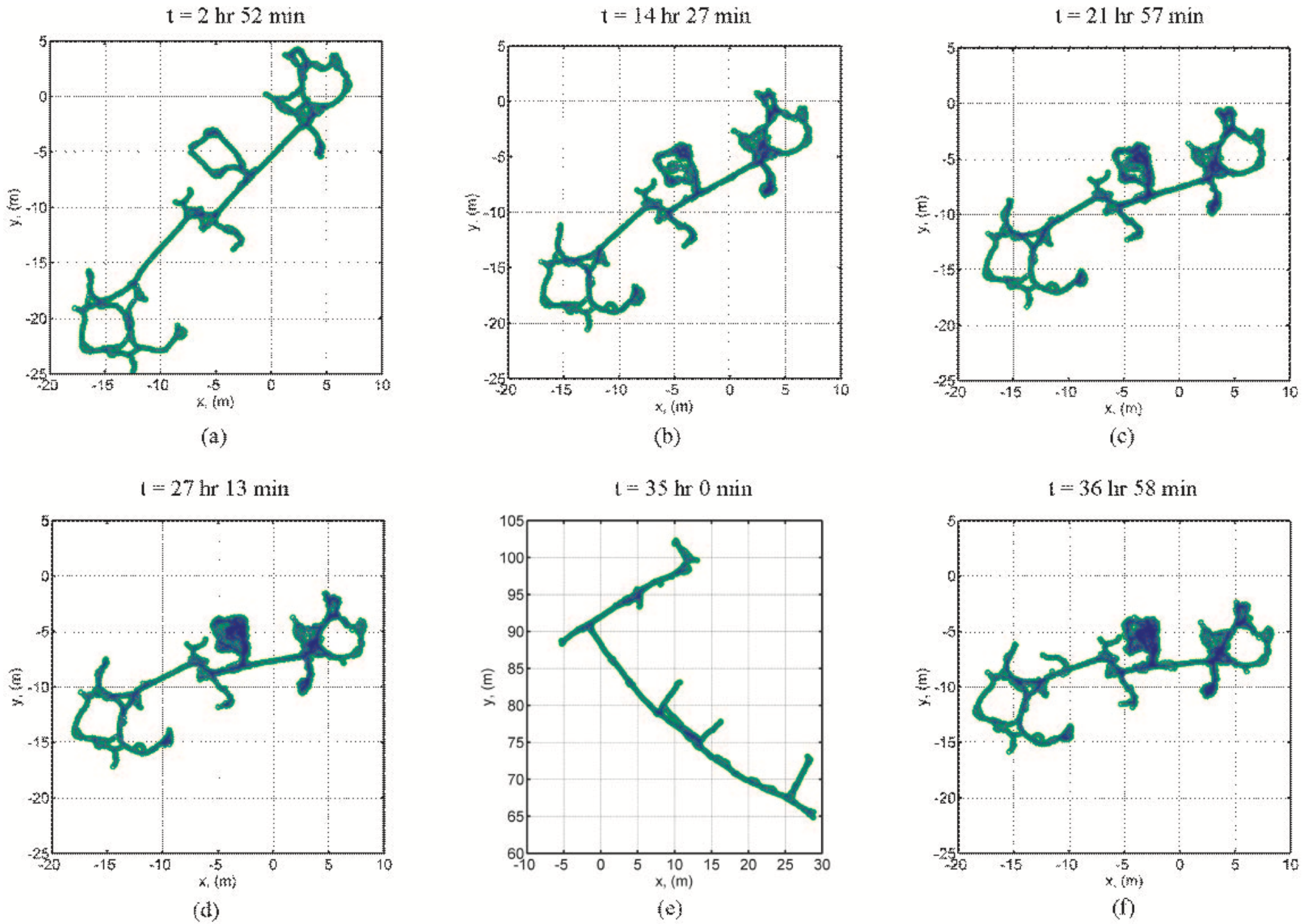
- Robot navigation and SLAM based on computational models of the hippocampus
- Cognitive map mechanisms
  - appearance based visual scene matching
  - competitive attractor networks
  - semi-metric topological map representation
- Components
  - Pose cells: analogous to rat’s grid cells
  - Local view cells: interface to the robot’s sensors
  - Experience map: functionally replaces place cells

Milford & Wyeth (2010). Persistent Navigation and Mapping Using a Biologically Inspired Slam System. IJRR, 29, 9 , 1131–53

RatSLAM



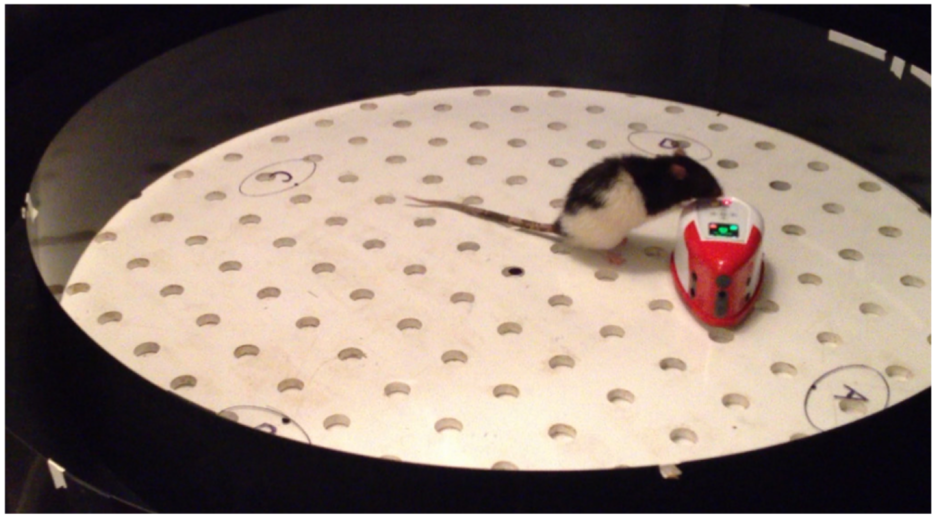
Milford & Wyeth 2010



Milford & Wyeth 2010

RatSLAM: Rat meets iRat

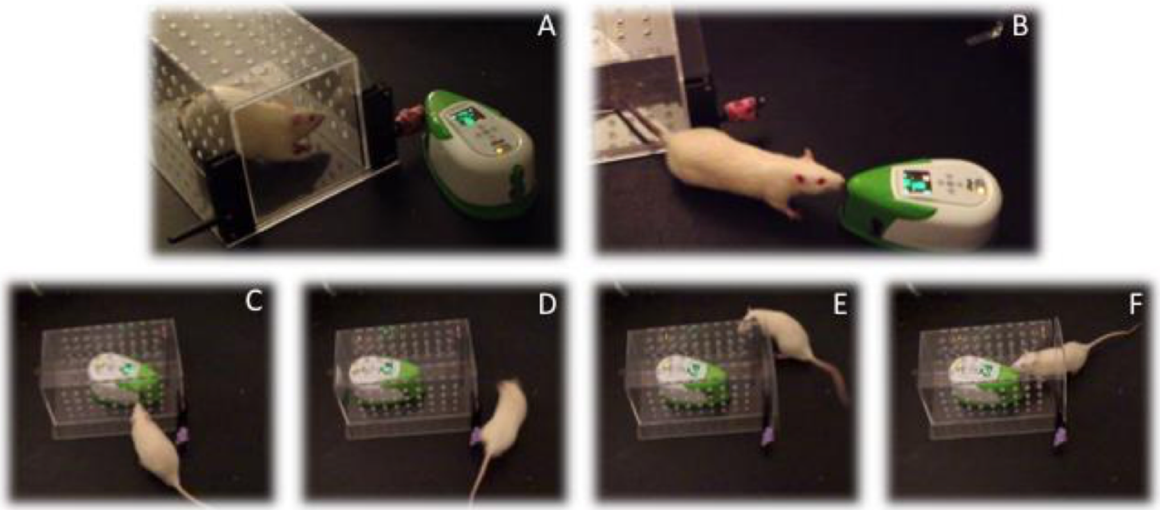
- iRat: rat-like robot for spatial navigation experiments (Wiles' Lab)
  - Application of RatSLAM to mobile (iRat) robots
  - Evolution of language in robots (Lingodroids)
  - **Experiments with real rats!**



Wiles et al. (2012). Rat meets iRat. IEEE ICDL-EpiRob Quinn et al. (2018). When rats rescue robots. ABC journal

RatSLAM: Rat meets iRat

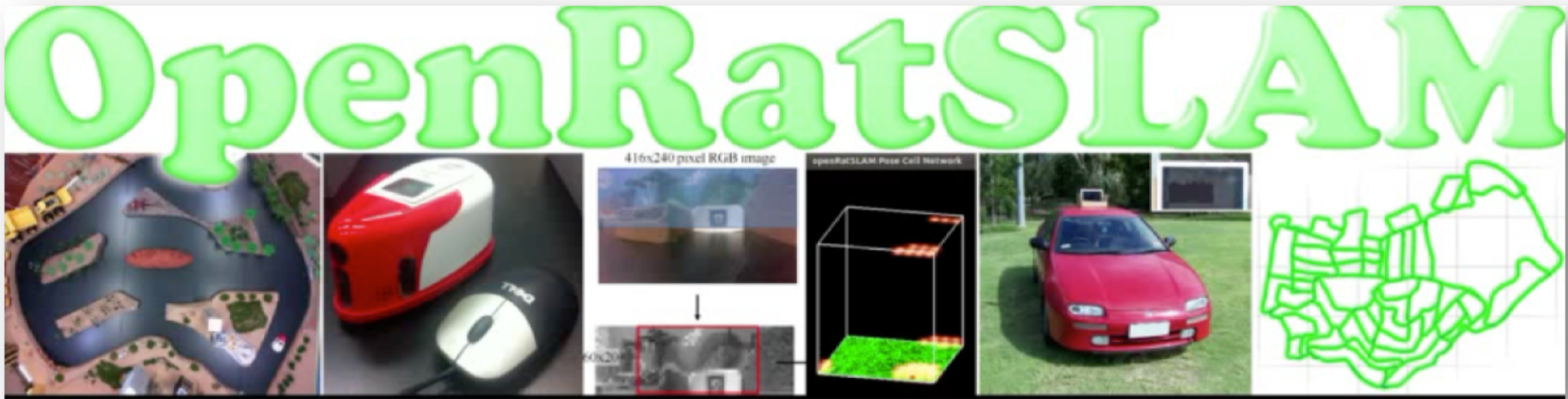
- iRat: rat-like robot for spatial navigation experiments (Wiles' Lab)
  - Application of RatSLAM to mobile (iRat) robots
  - Evolution of language in robots (Lingodroids)
  - Experiments with real rats!
  - **Rats helping robots!!**



Wiles et al. (2012). Rat meets iRat. IEEE ICDL-EpiRob Quinn et al. (2018). When rats rescue robots. ABC journal

OpenRatSLAM

- Open-source version of RatSLAM with bindings to the Robot Operating System (ROS)
  - Interconnected ROS nodes for pose cells, experience map, local view cells, and visual odometry estimates
  - three publicly available open-source datasets

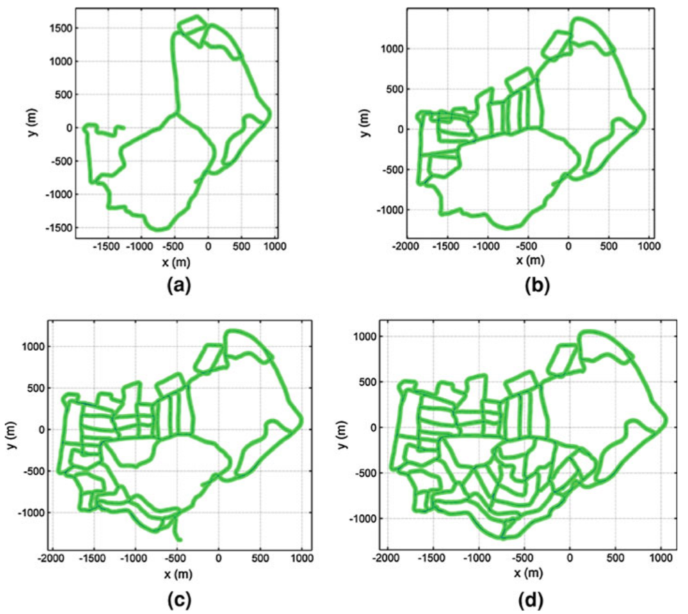


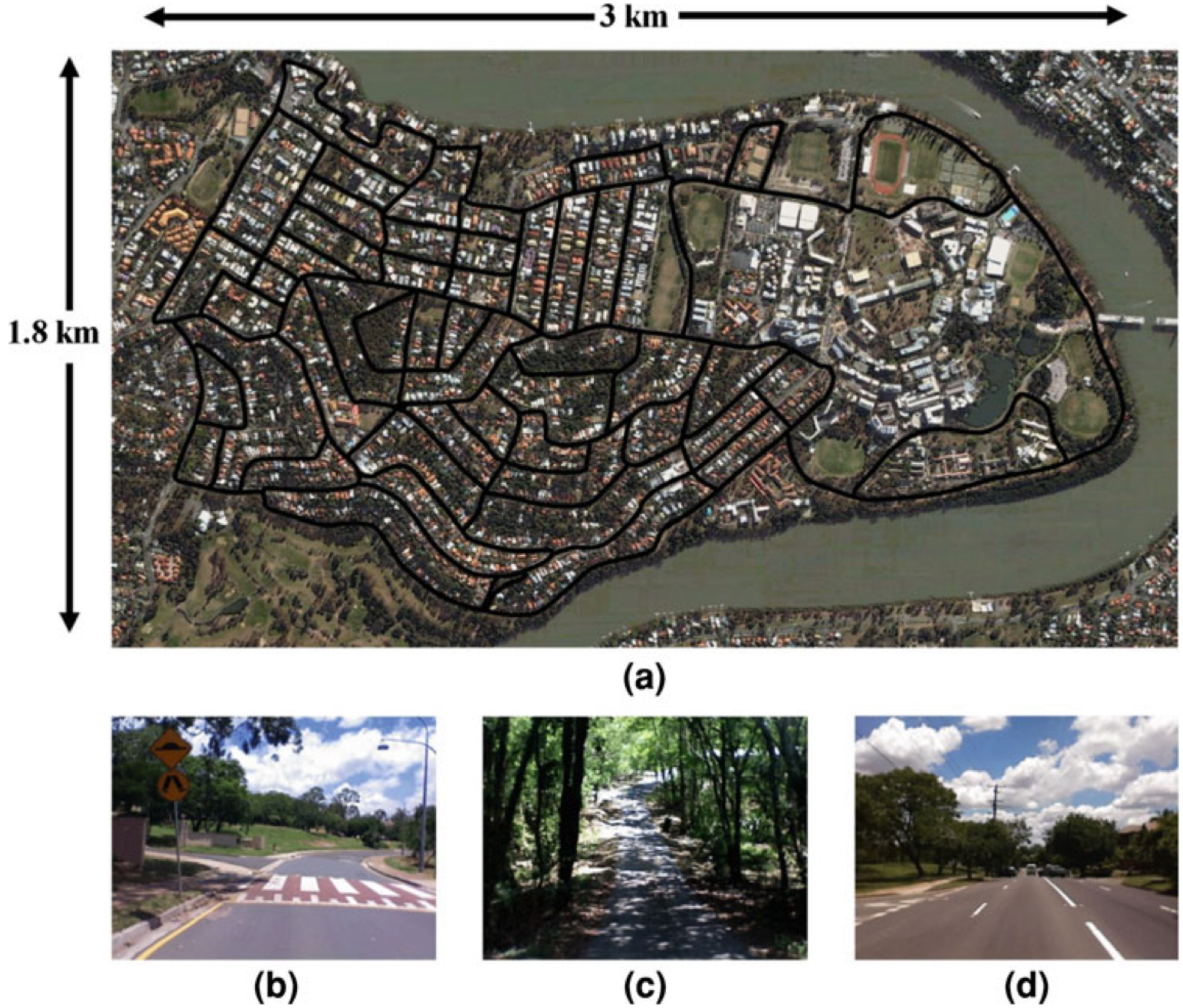
http://github.com/coxlab/ratslam-python

Ball et al. (2013) OpenRatSLAM. Autonomous Robots

* OpenRatSLAM

- St Lucia suburb
  (66km car
  journey)



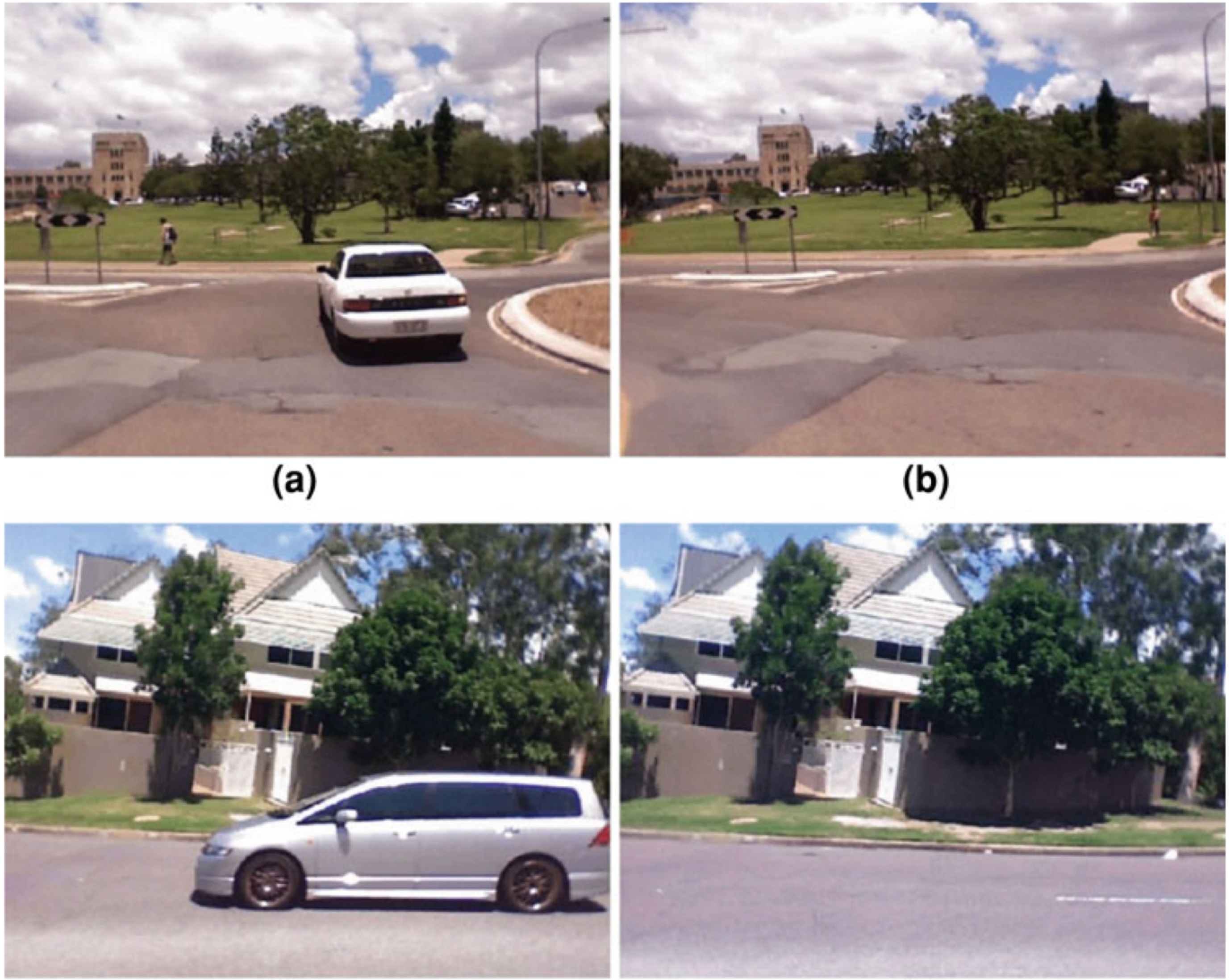


http://github.com/coxlab/ratslam-python

Ball et al. (2013) OpenRatSLAM. Autonomous Robots

* OpenRatSLAM

- St Lucia Suburb
  (66km car journey)



(a) (b)

(c) (d)

http://github.com/coxlab/ratslam-python

Ball et al. (2013) OpenRatSLAM. Autonomous Robots

* OpenRatSLAM

www.youtube.com/watch?v=p8mf0SZKneY

Summary

- Navigation and Manipulation
- Path planning
- Simultaneous Localisation and Mapping
  - SLAM
  - vSLAM
- Cognitive SLAM: RatSLAM

- Reading (optional)
  - Riisgaard & Blas: SLAM for Dummies
  - Ball et al. (2013) OpenRatSLAM