

Building and managing maps for autonomous robots

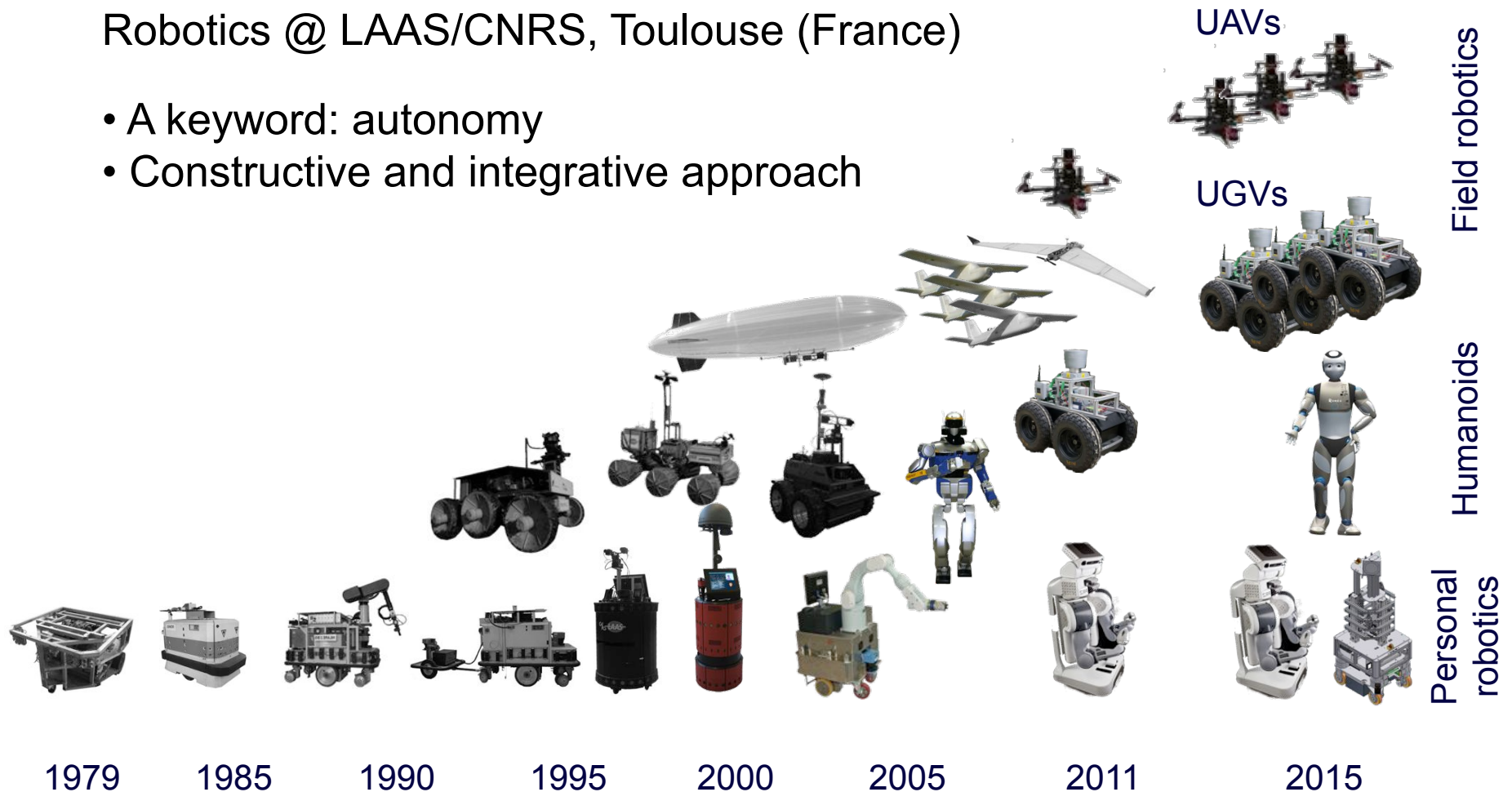
Simon Lacroix
*Laboratory for Analysis
and Architecture of Systems*
CNRS, Toulouse



Where do I come from?

Robotics @ LAAS/CNRS, Toulouse (France)

- A keyword: autonomy
- Constructive and integrative approach



Mobile robots everywhere



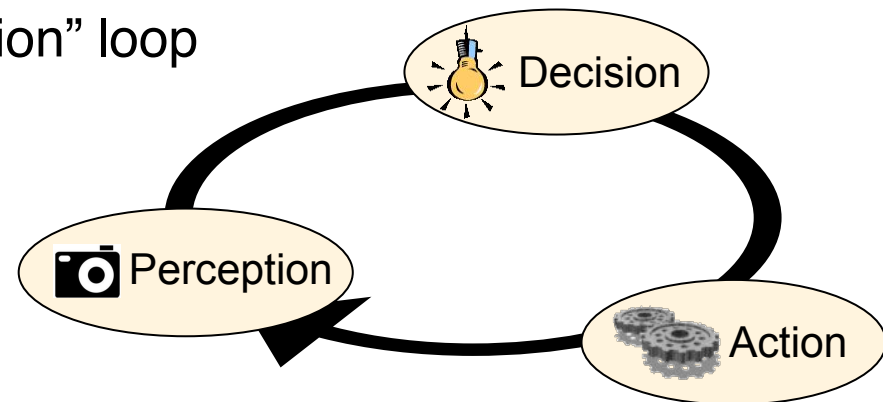
From automatic control to autonomous control

- Automatic control :
 - Well defined task (*“regulate variable”, “follow trajectory”...*)
 - Well defined and simple environments
 - “Direct” link between (simple) perception and action
- Autonomous control :
 - More general task (*“reach goal”, “monitor area ”...*)
 - More complex environments
 - Calls for *decisional processes*

⇒ “Perception / Decision / Action” loop

Plus :

- Processes *integration*
- Learning
- Interaction with humans
- Interactions with other robots
- ...



From automatic control to autonomous control

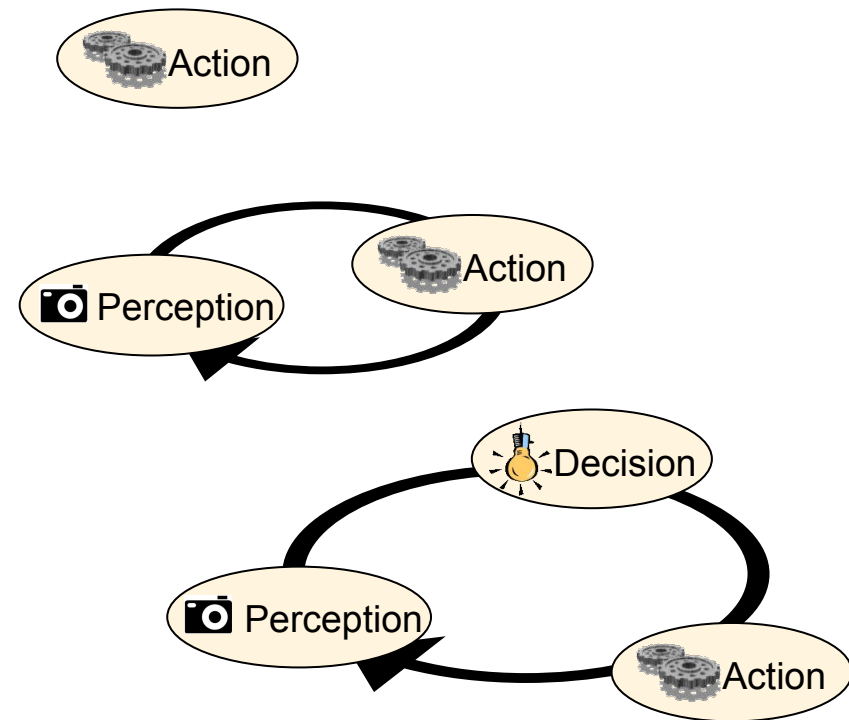
E.g. for a drone:

- Regulate heading / speed / altitude
- Follow a list ordered waypoints
- Follow a geometric trajectory

- Follow a road
- Follow a target

- Survey an area while avoiding threats and obstacles, and track intruders

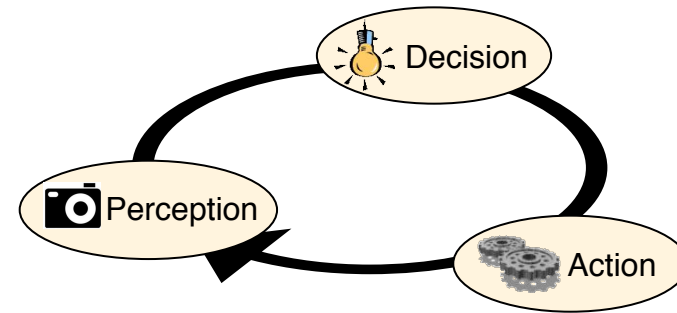
*Decision: notion of deliberation,
planning: prediction and evaluation
of the outcomes of an action*



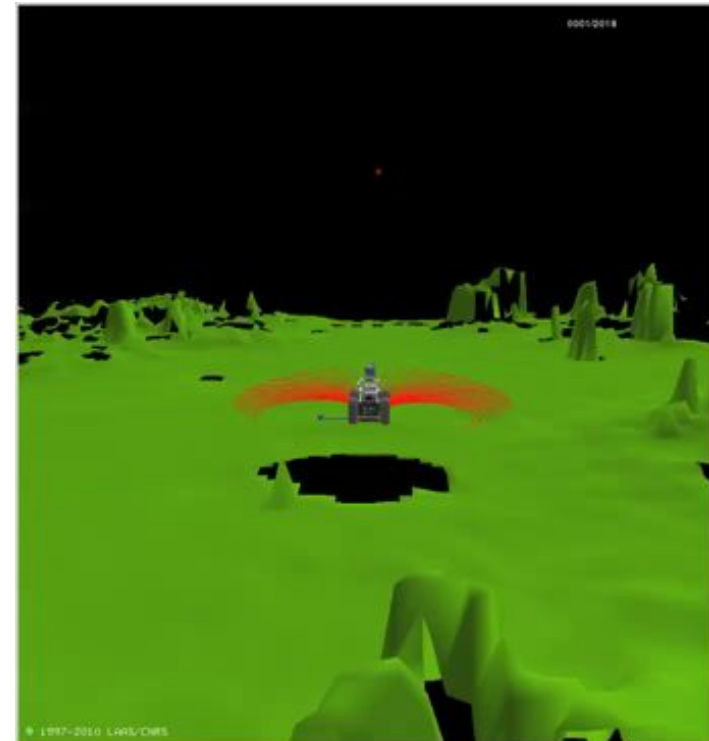
Also referred to as “Sense/
Plan/Act” paradigm
(this scheme applies to
numerous processes)

Autonomous robots

- The three pillars of autonomy

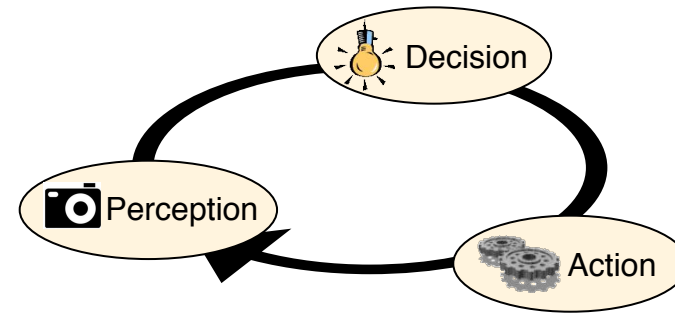


- A basic task: autonomous navigation



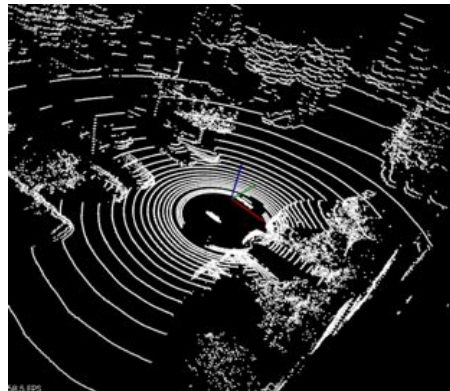
Autonomous robots

- The three pillars of autonomy

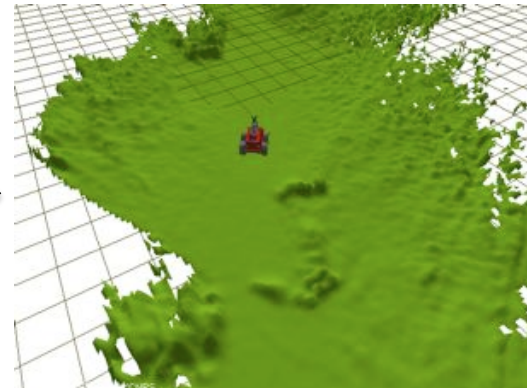


- A basic task: autonomous navigation

1. Perception



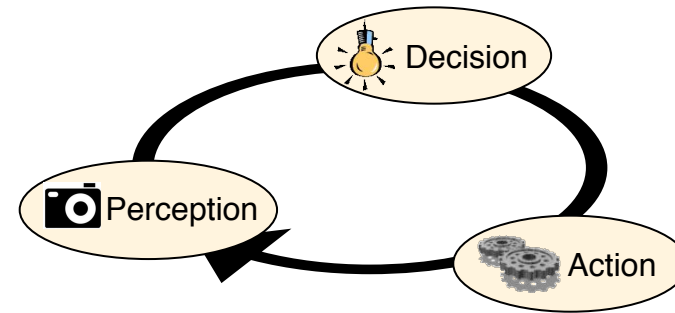
Depth image



Digital terrain model

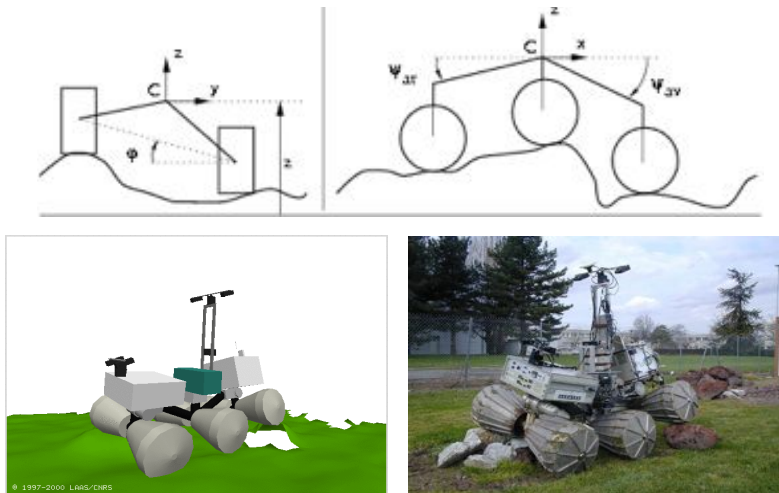
Autonomous robots

- The three pillars of autonomy

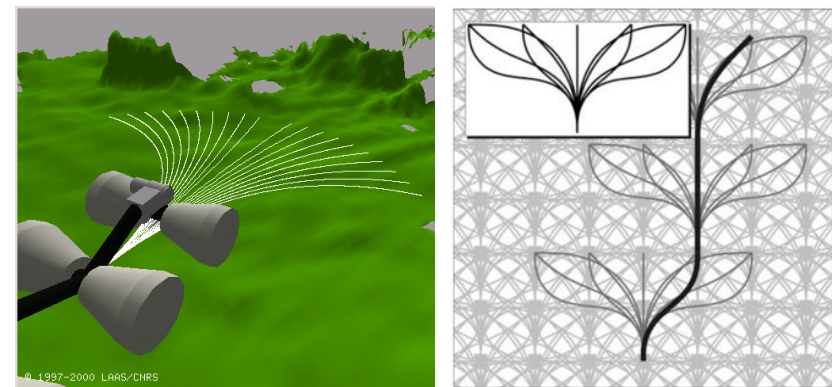


- A basic task: autonomous navigation

2. Decision



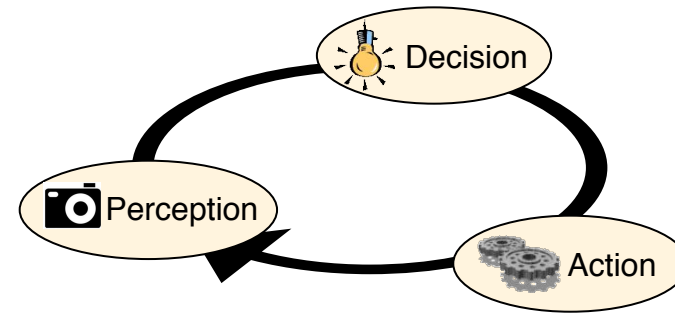
Convolution of the robot model with the terrain model



Search

Autonomous robots

- The three pillars of autonomy



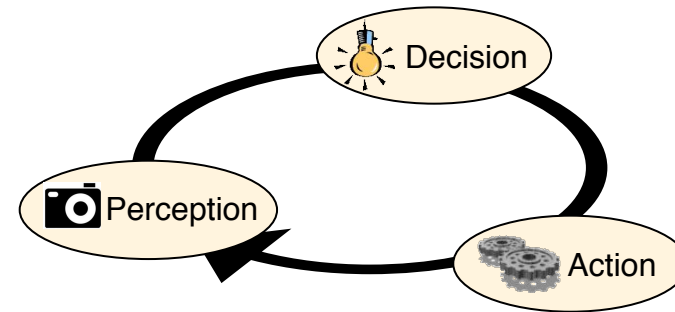
- A basic task: autonomous navigation

3. Action

“Just” execute the planned motions

Autonomous robots

- The three pillars of autonomy



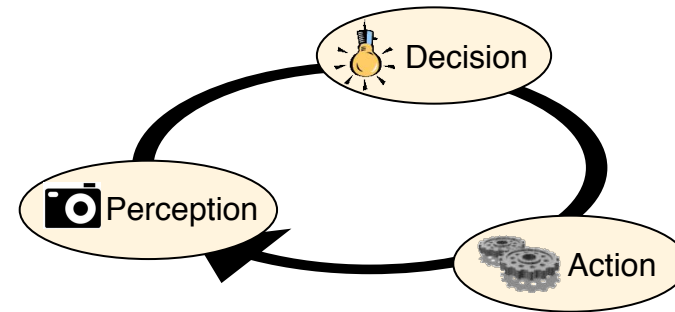
2. Decision

Deciding, Planning = Simulation + Search

- Simulation of the effects of an action with a predictive model
- Search over possible organizations of possible actions to meet a goal or to optimize a criteria

Autonomous robots

- The three pillars of autonomy



2. Decision

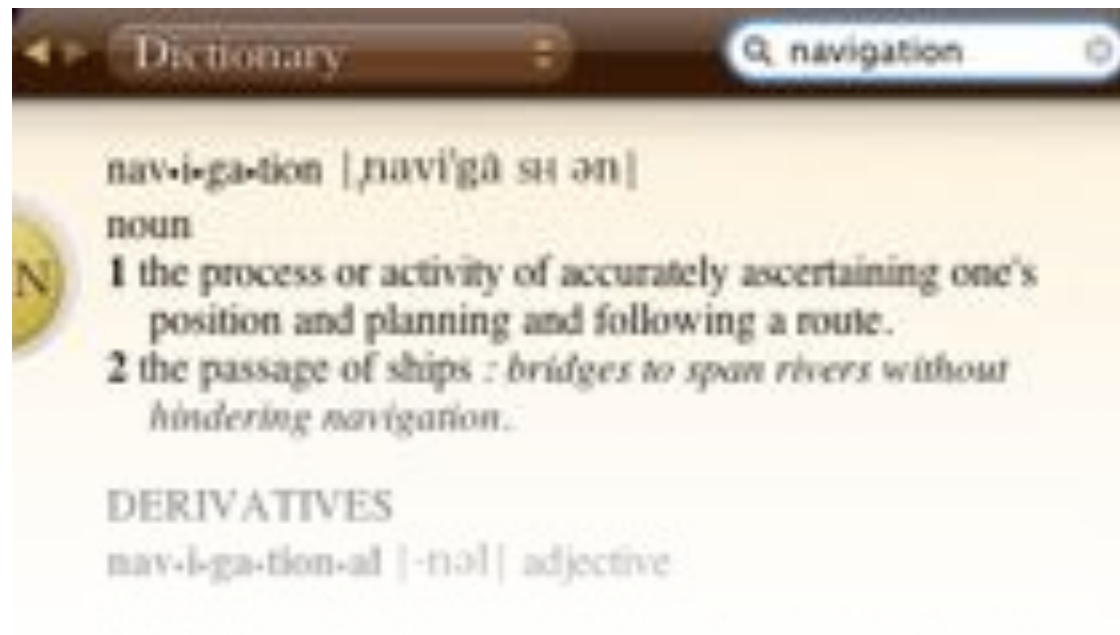
➔ Models are at the core of autonomy

- Environment models
- Action models

Outline

- Autonomous robots
- Mapping for autonomous navigation
- Mapping and localization
- Beyond autonomous navigation

Autonomous navigation

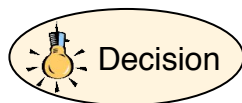


Autonomous navigation

An instance of the perception/decision/action loop:



- Gathering data on the environment
- Structuring the data into *models (maps)*
- Localizing the robot



- Planning motions to find the optimal one



- Executing a trajectory

(Autonomous navigation is declined in numerous flavors)

Depending on:

- The kind of vehicle
 - Air, ground, maritime, underwater
- The kind of environment
 - Structured or not
 - “Hostile” or not
 - Dynamic or not
 - ...
- Numerous constraints or helpers
 - Environment known or not?
 - Existence of localisation aiding systems?
 - Networked environment or not?
 - ...



Mapping for autonomous navigation

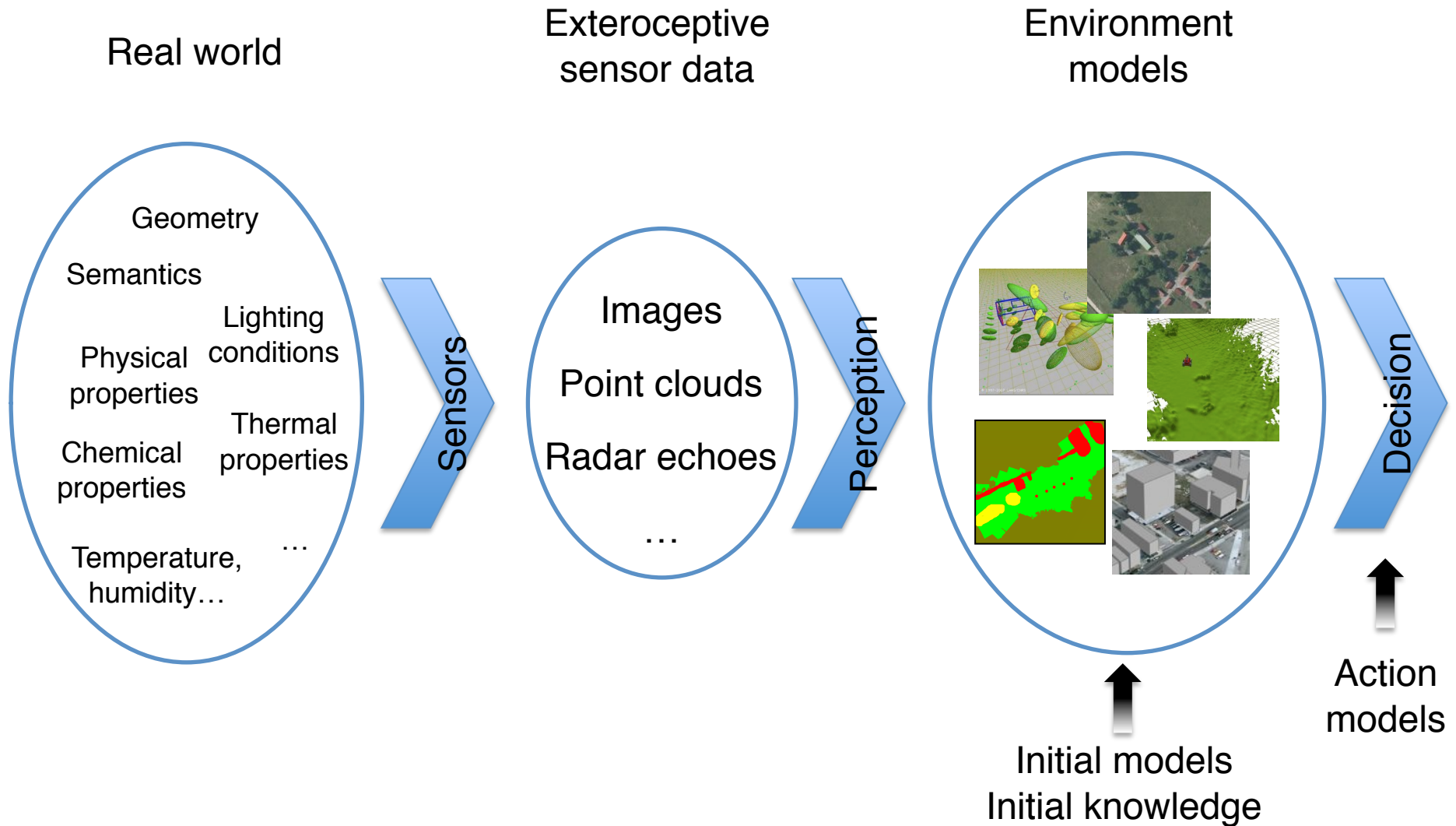
Autonomous navigation involves:

- Avoiding obstacles
- Planning trajectories
- Planning itineraries

↳ Need to detect and represent obstacles
Need to estimate feasibility / cost / time of motions

These are assessed on the basis of *maps*

Building maps: information flow



Mapping for autonomous navigation

Mapping = “Acquisition and representation of information on the environment”

All sensors are *imprecise* and *uncertain*

↳ Probabilistic mapping:

Knowing perfectly the robot poses $\{u_i\}$, and given the sensor data $\{z_i\}$:

$$d = \{u_1, z_1, u_2, z_2, \dots, u_n, z_n\}$$

calculate the most likely map:

$$m^* = \arg \max_m P(m | d)$$

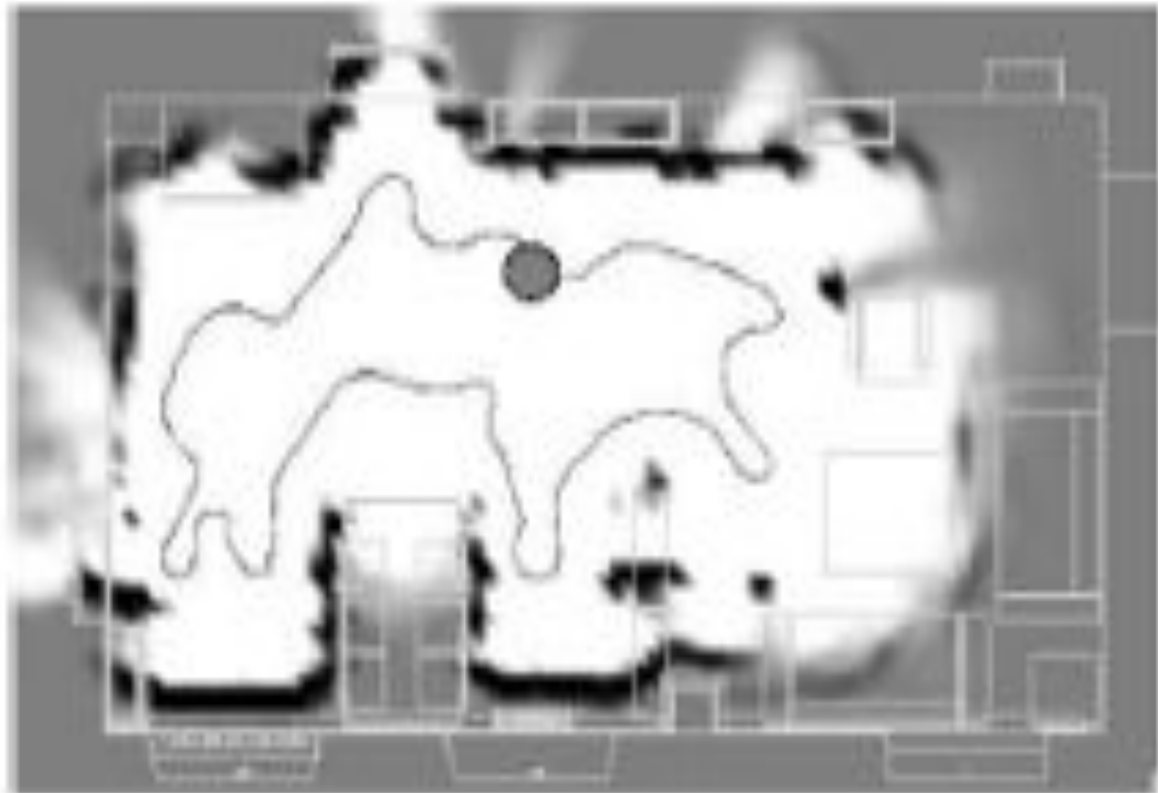
Probabilistic mapping

Illustration: occupancy grid

- Structure: 2D regular discretization of the environment (= 2D grid)
- Each grid cell encodes the probability of presence of an obstacle (“occupied”)

$$P(obst) = 1$$

$$P(obst) = 0$$



Probabilistic mapping

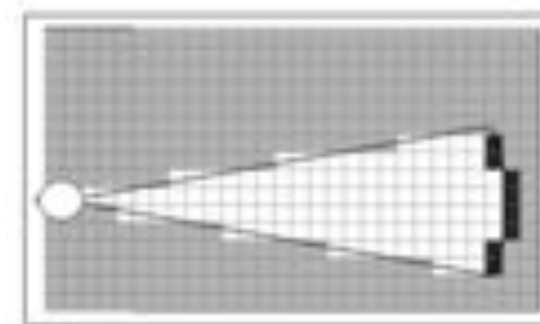
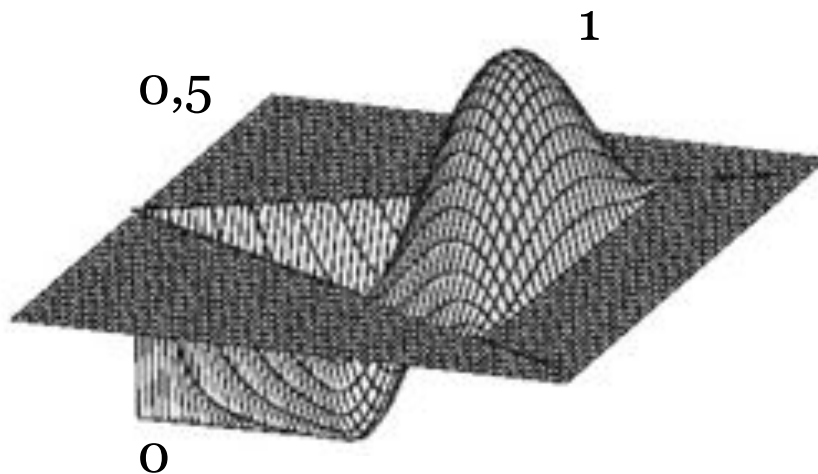
Illustration: occupancy grid

- Bayes theorem:

Sensor model Initial knowledge

$$P(O_i|z) = \frac{P(z|O_i)P(O_i)}{P(z)}$$

- Sensor model: probability of range reading given known occupancy of a grid cell O_i :
- “Inverse sensor model”

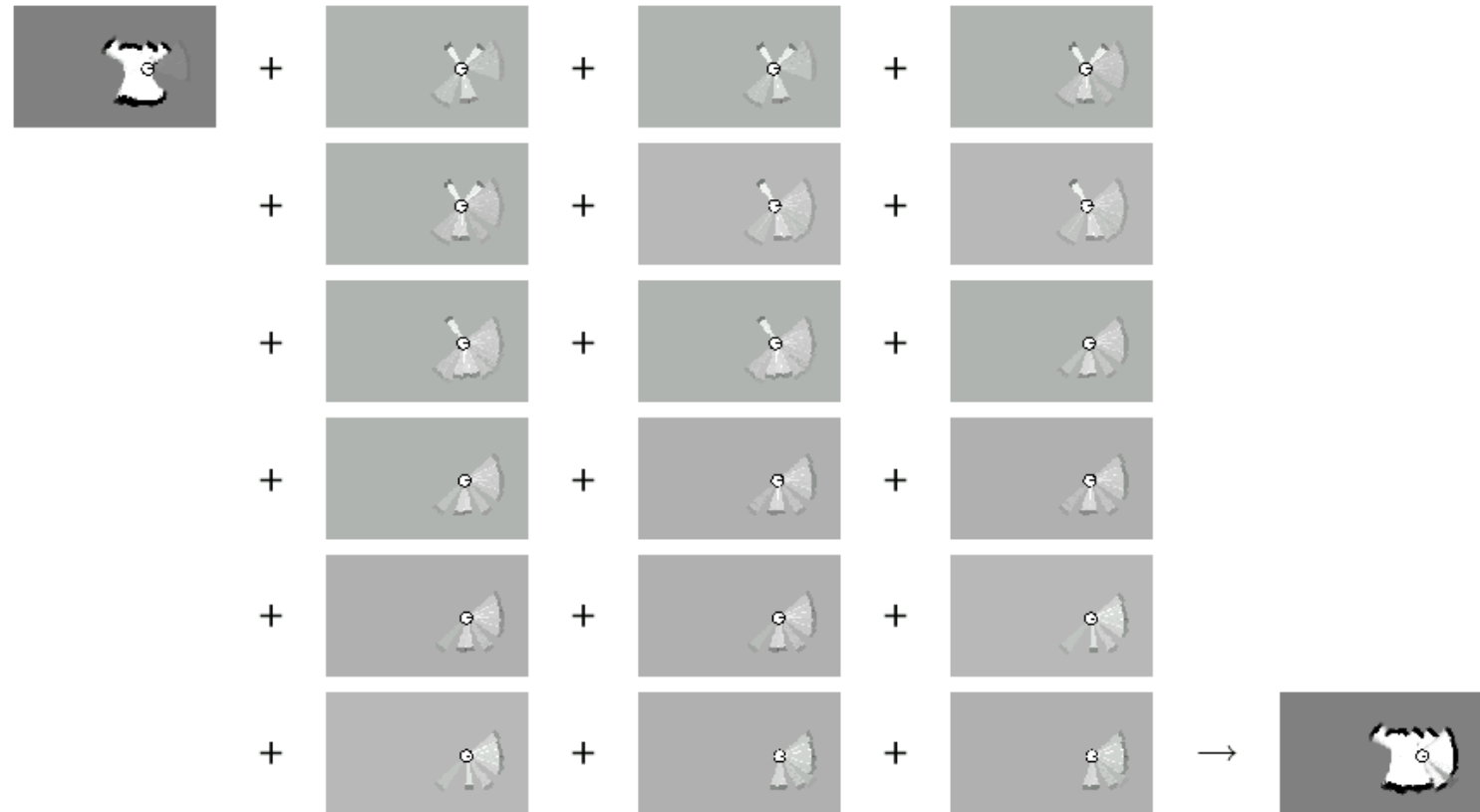


Sonar sensor

Probabilistic mapping

Illustration: occupancy grid

- Recursive bayes updates



Probabilistic mapping

Illustration: occupancy grid



Sonar



Planar Lidar

Probabilistic mapping

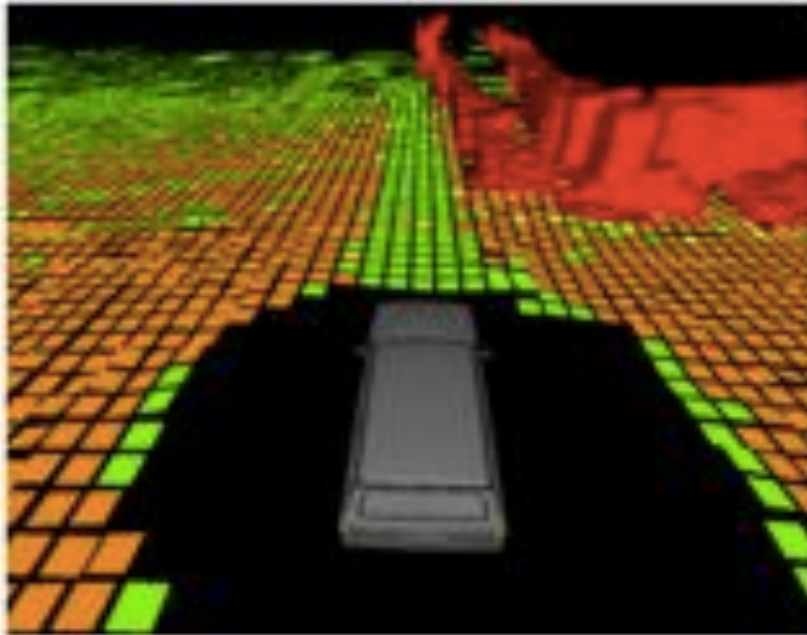
Illustration: terrain navigability using point clouds



Probabilistic mapping

Illustration: terrain navigability using point clouds

- Same principles apply: modeling the sensor, application of the Bayes theorem, recursive update
- More complex sensors: use of learning techniques to establish their model

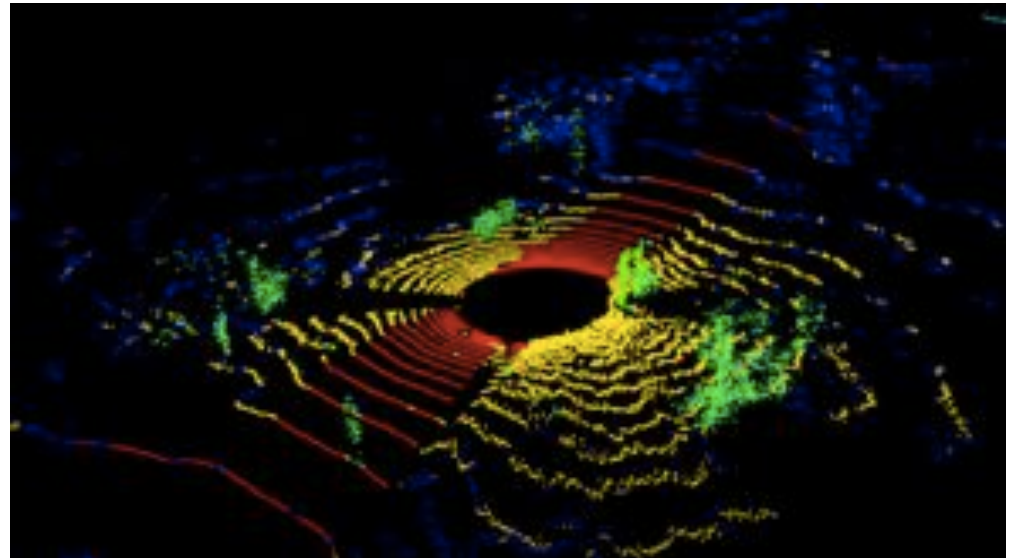


Data classification – area-wise

Probabilistic mapping

Illustration: terrain navigability using point clouds

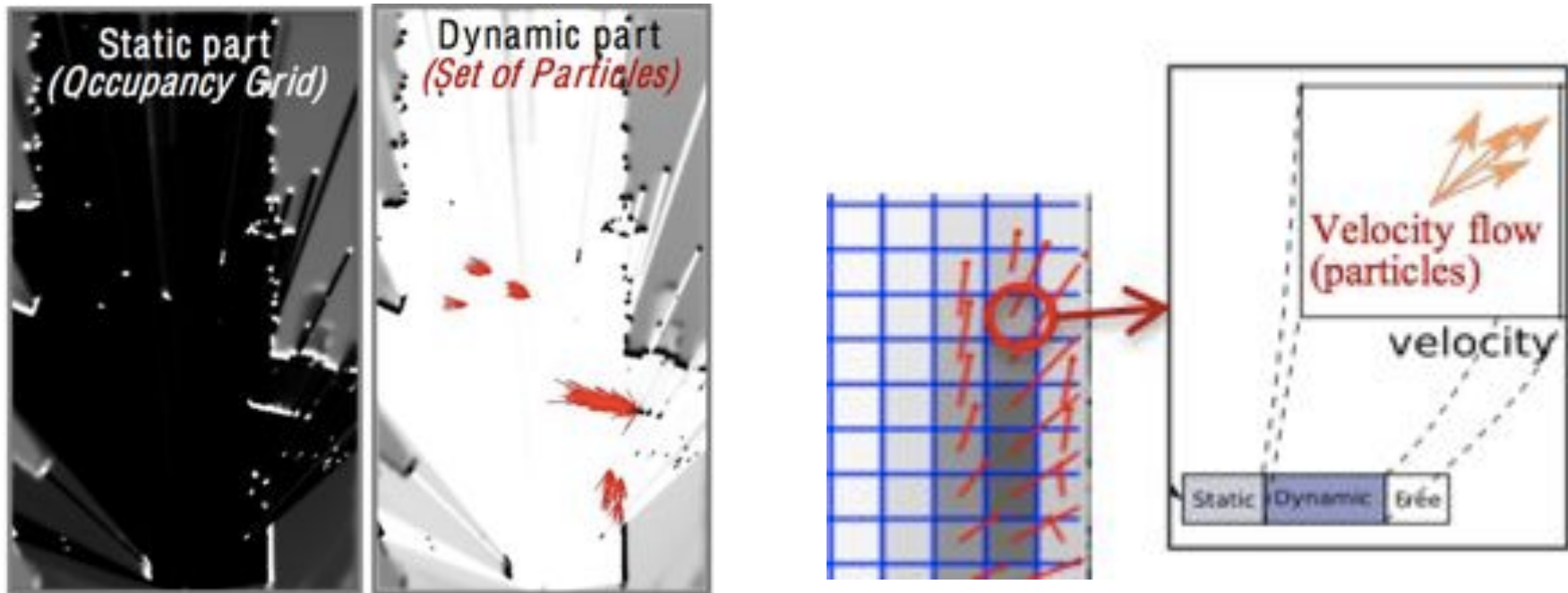
- Same principles apply: modeling the sensor, application of the Bayes theorem, recursive update
- More complex sensors: use of learning techniques to establish their model



Data classification – point-wise

Probabilistic mapping

Illustration: mapping dynamic environments

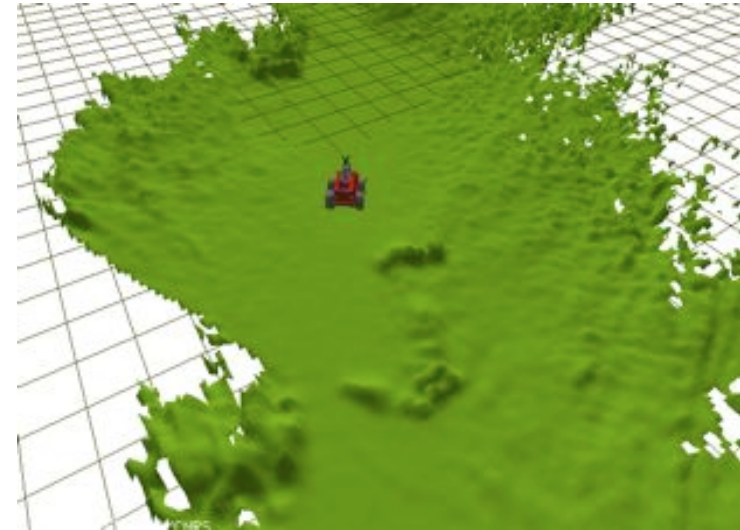


(INRIA eMotion)

Geometric mapping

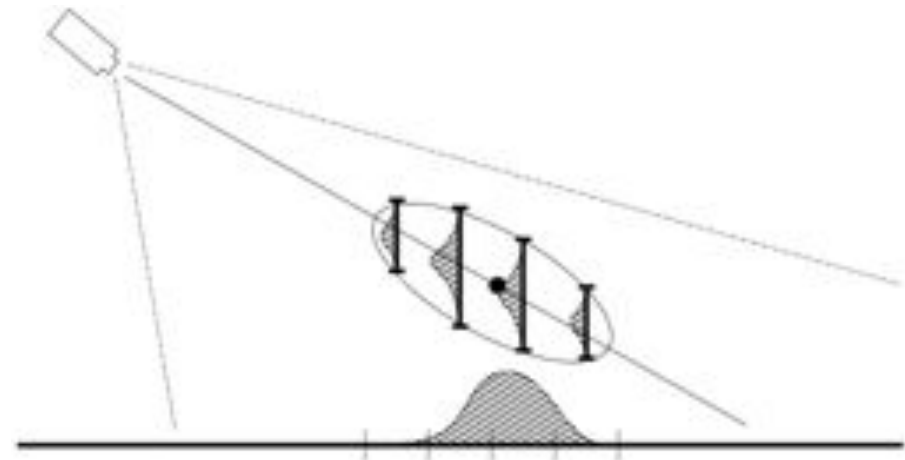
Illustration: digital terrain maps

DTM: $z = f(x, y)$ on a regular Cartesian grid



- Varying resolution
- Imprecision on the data

↳ uncertainties in the values

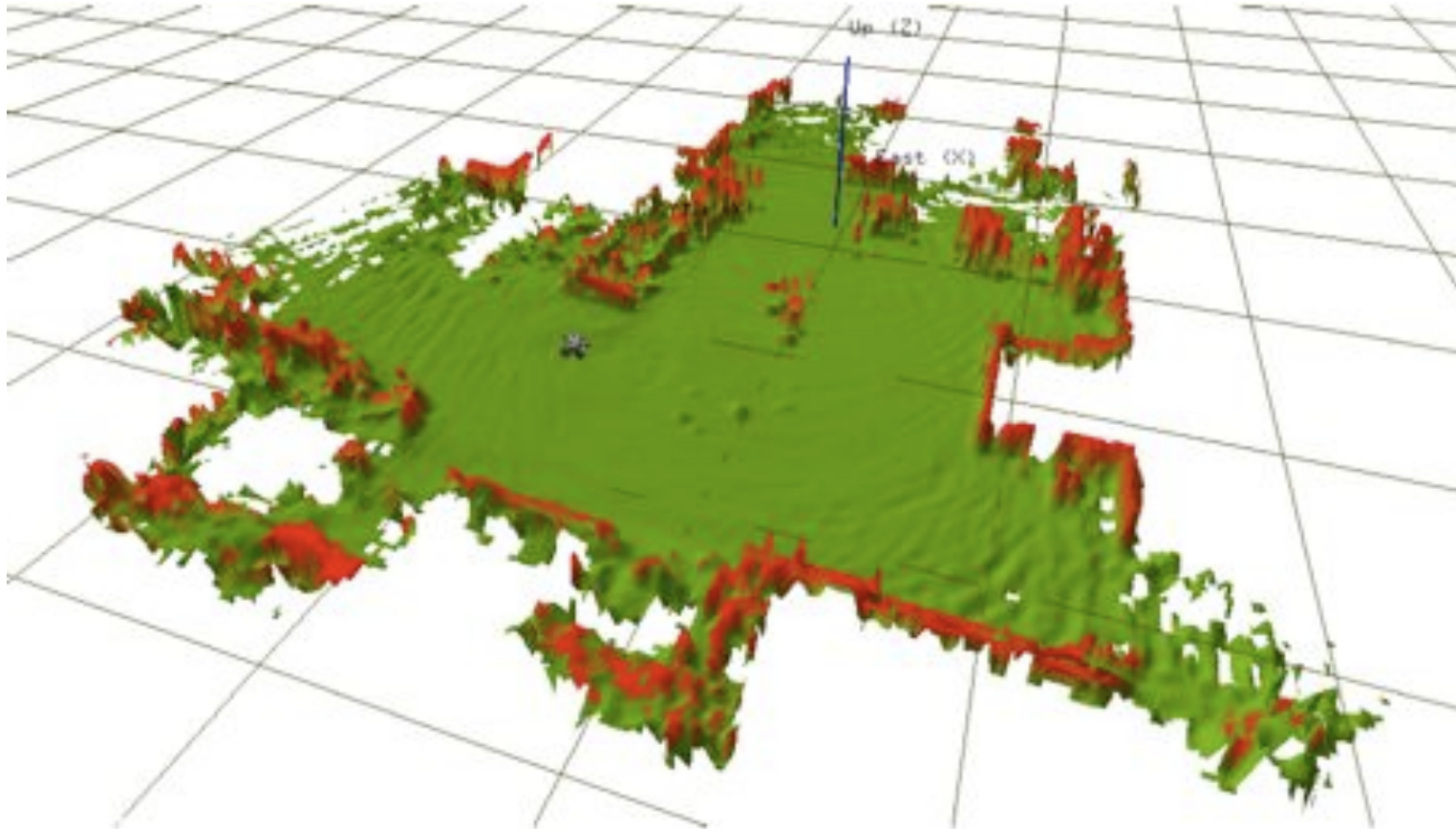


Geometric mapping

Illustration: digital terrain maps

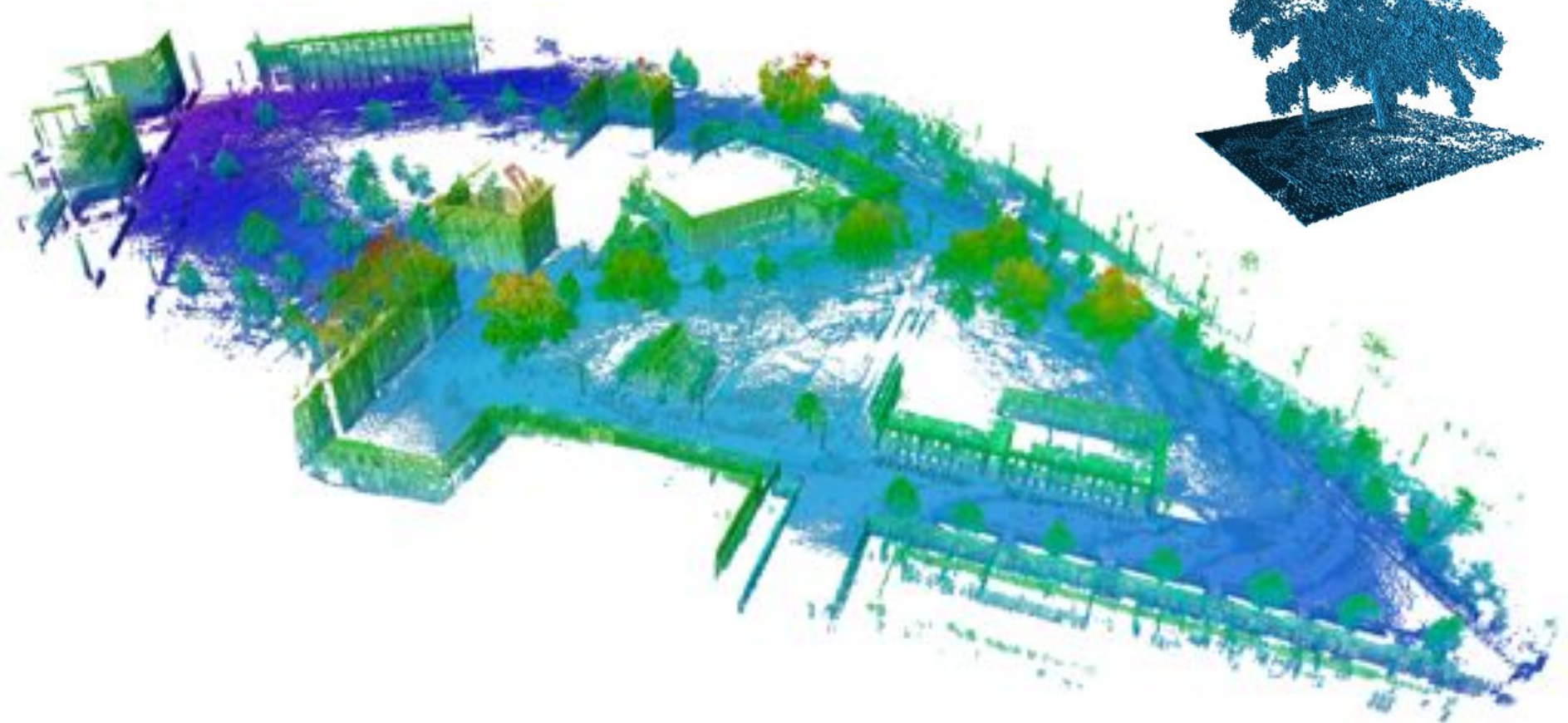
Practical implementation:

- simple statistics on the “population” of 3D points that fall in a cell
- a cell is : (z, σ_z) (+ #points, dates, ...)



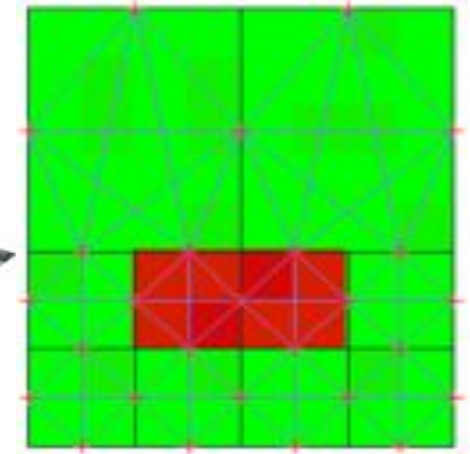
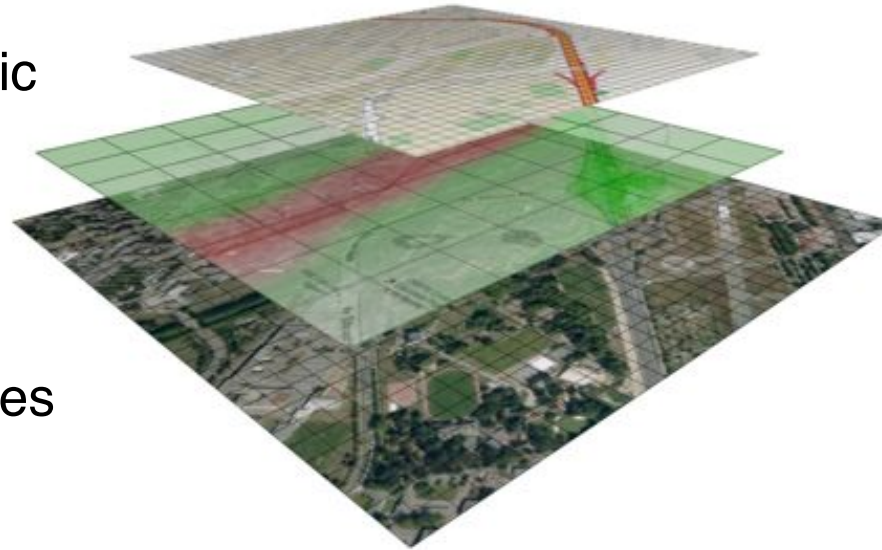
Geometric mapping

Illustration : approximate hierarchical 3D occupancy grids (“Octomap”)



Map structures

- Raster models:
regular or hierarchic
Cartesian grids



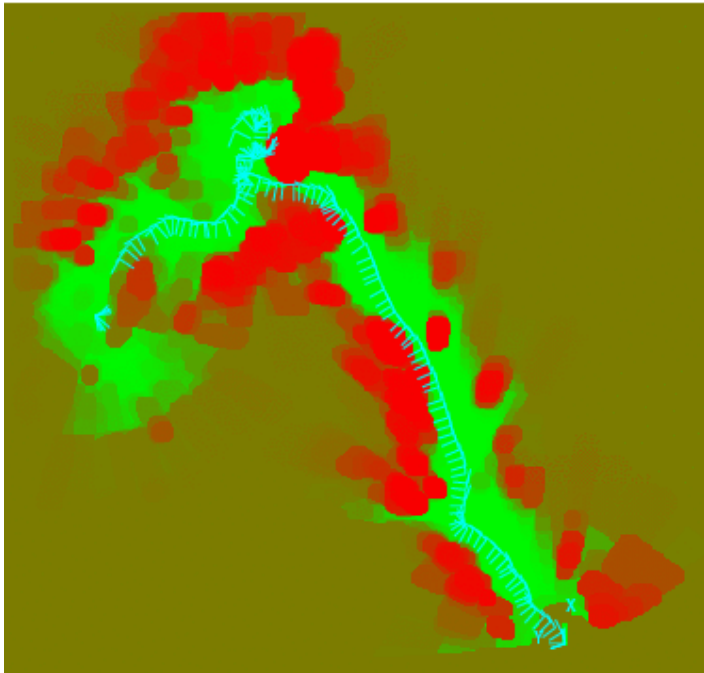
➔ Graph structures
easily derived

- Real 3D structures are
hardly maintained in
real time

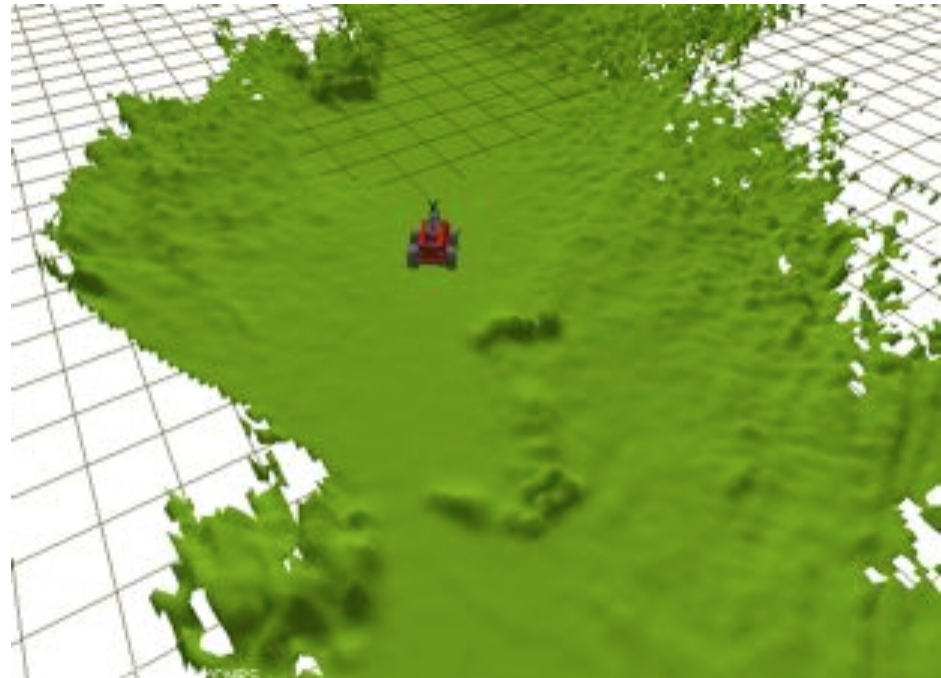


Exploiting maps for navigation

- To generate/plan motions
 - ⇒ evaluation/quantification of motion actions
- Classical 2-stages approach: itineraries / trajectories



Probabilistic obstacle model



Digital elevation map

Maps for navigation: key points

1. Whatever the encoded information (terrain class, elevation, traversability, ...), it is *essential* maintain its “quality” (confidence, precision, certainty...):
 - To fuse the various sources of information
 - initial model
 - models built by other robots
 - sensor data
 - To drive the decision processes
2. Spatial consistency is crucial

Mapping for autonomous navigation

All sensors are *imprecise* and *uncertain*

↳ Probabilistic mapping:

Knowing perfectly the robot poses $\{u_i\}$, and given the sensor data $\{z_i\}$:

$$d = \{u_1, z_1, u_2, z_2, \dots, u_n, z_n\}$$

calculate the most likely map:

$$m^* = \arg \max_m P(m | d)$$

Mapping for autonomous navigation

All sensors are *imprecise* and *uncertain*



Probabilistic mapping:

???

Knowing perfectly the robot poses $\{u_i\}$, and given the sensor data $\{z_i\}$:

$$d = \{u_1, z_1, u_2, z_2, \dots, u_n, z_n\}$$

calculate the most likely map:

$$m^* = \arg \max_m P(m | d)$$

Outline

- Autonomous robots
- Mapping for autonomous navigation
- **Mapping and localization**

On the importance of localization

Autonomous navigation encompasses:

- Ensuring the lowest level (locomotion) controls
- Ensuring the proper execution of paths / trajectories
- Ensuring the achievement of the navigation mission (“goto [goal]”)
- Ensuring the spatial consistency of the built models

AMERICA.



Mapping and localization

Mapping as an estimation problem:

Given the robot position x_v , a data set Z_k , a sensor model $p(\mathbf{Z}^k | \mathbf{M}, \mathbf{x}_v)$ compute the map M : $p(\mathbf{M} | \mathbf{x}_v, \mathbf{Z}^k)$

Map-based localization as an estimation problem:

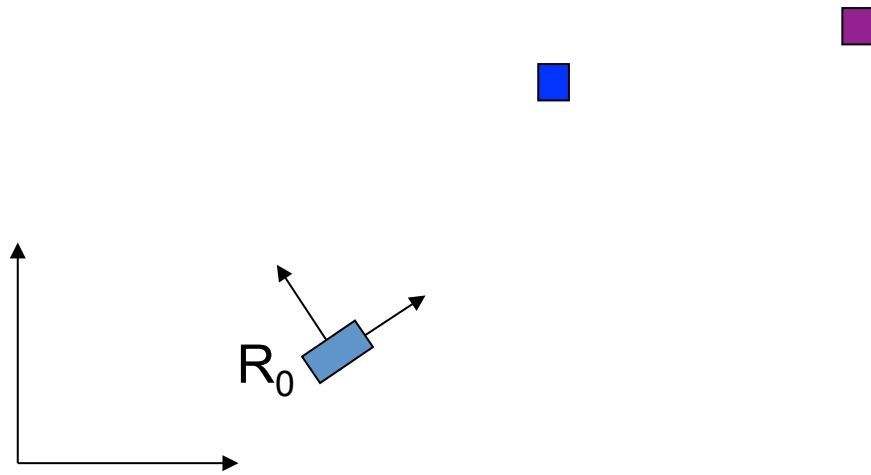
Given a map M , a data set Z_k , a sensor model $p(\mathbf{Z}^k | \mathbf{M}, \mathbf{x}_v)$ compute the robot position x_v : $p(\mathbf{x}_v | \mathbf{M}, \mathbf{Z}^k)$

In the absence of maps and precise localization:

Given the robot controls u_k and sensor readings z_k , compute the map M and the robot position x_v : $p(M, x_v | u_k, z_k)$

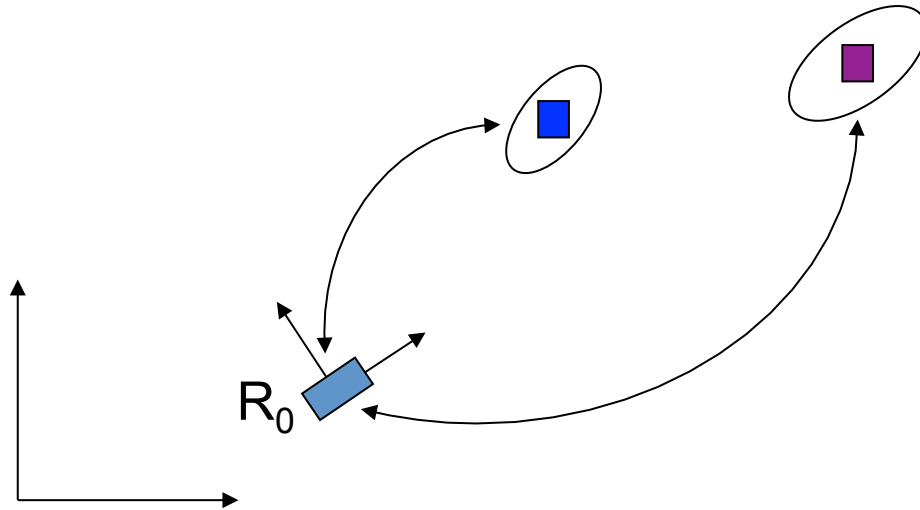
↳ Simultaneous Localization And Mapping (SLAM)

Principle of SLAM



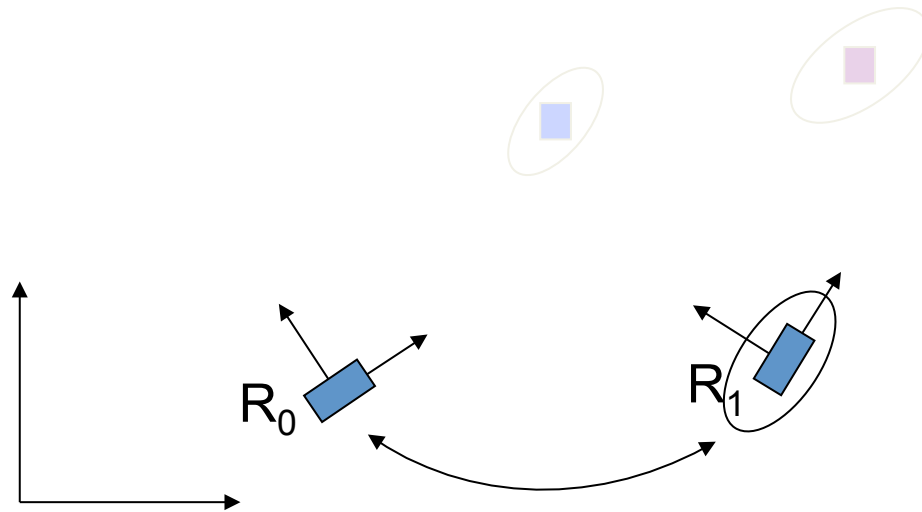
– Landmark detection

Principle of SLAM



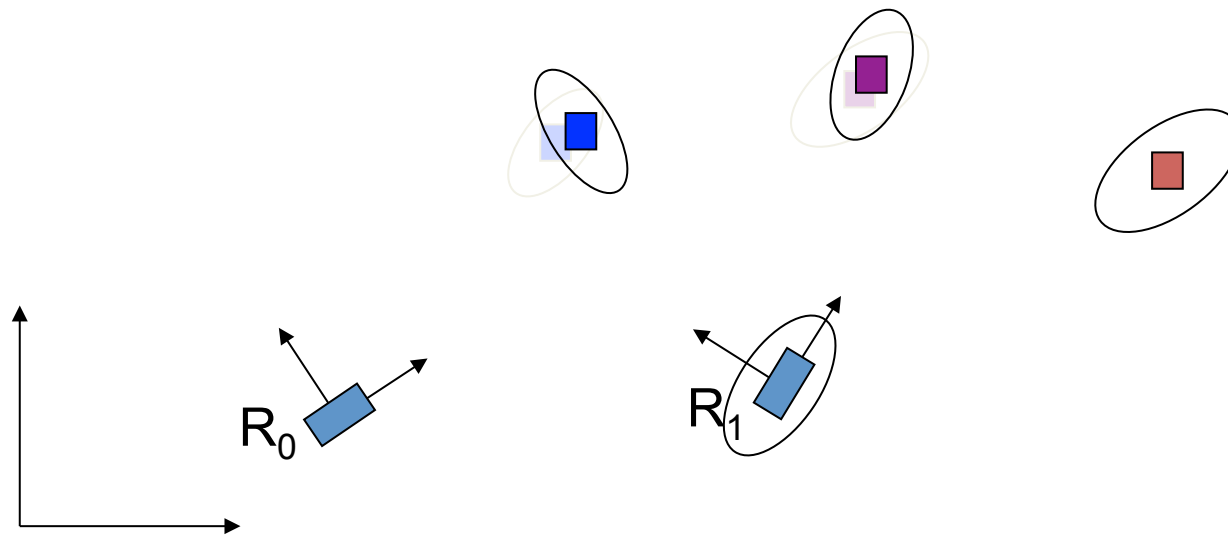
- Landmark detection
- Relative observations (measures)
 - Of the landmark positions

Principle of SLAM



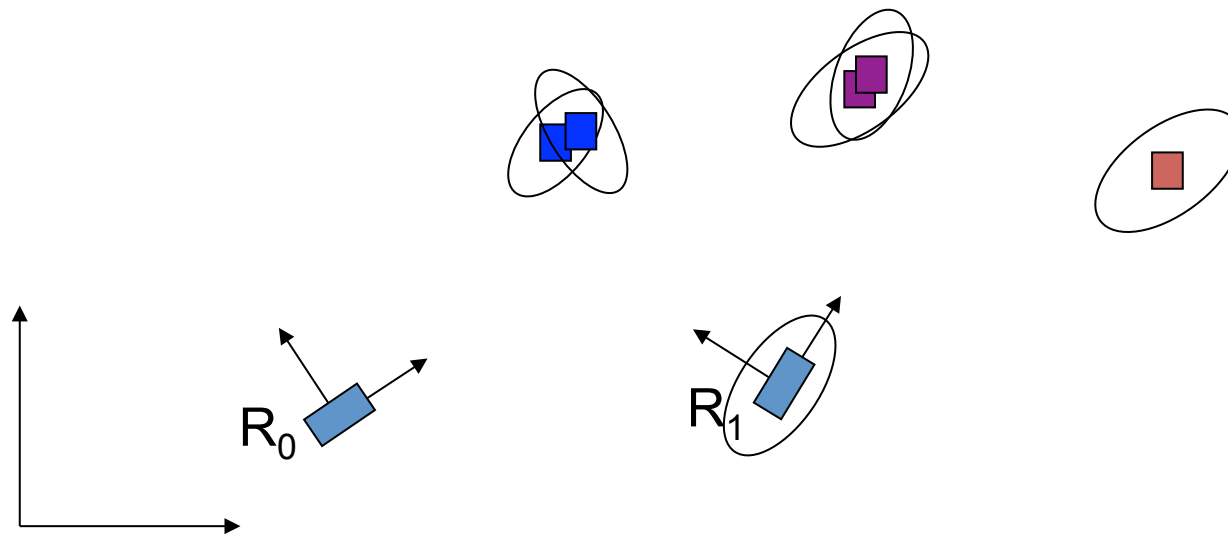
- Landmark detection
- Relative observations (measures)
 - Of the landmark positions
 - Of the robot motions

Principle of SLAM



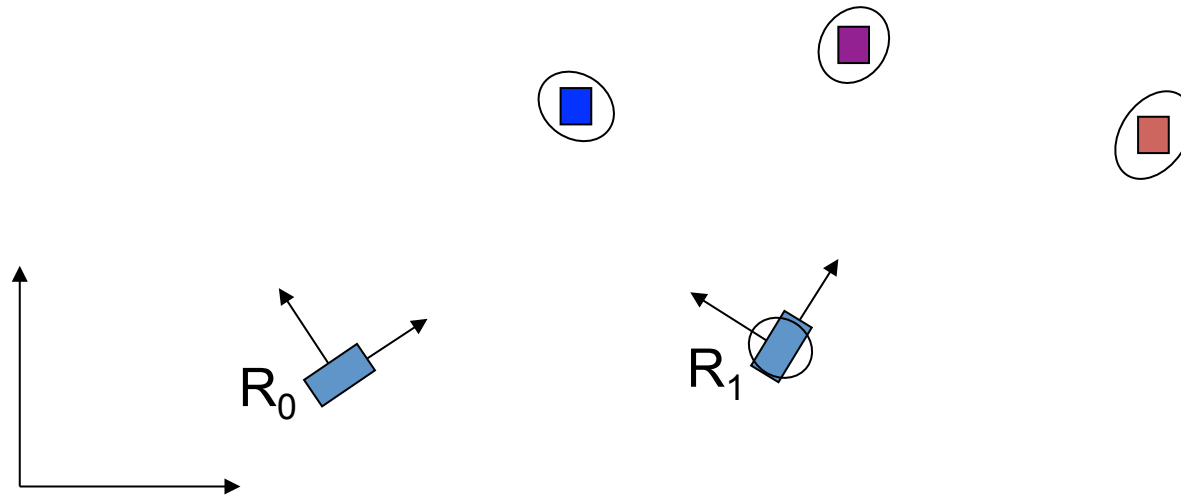
- Landmark detection
- Relative observations (measures)
 - Of the landmark positions
 - Of the robot motions

Principle of SLAM



- Landmark detection
- Relative observations (measures)
 - Of the landmark positions
 - Of the robot motions
- Observation associations

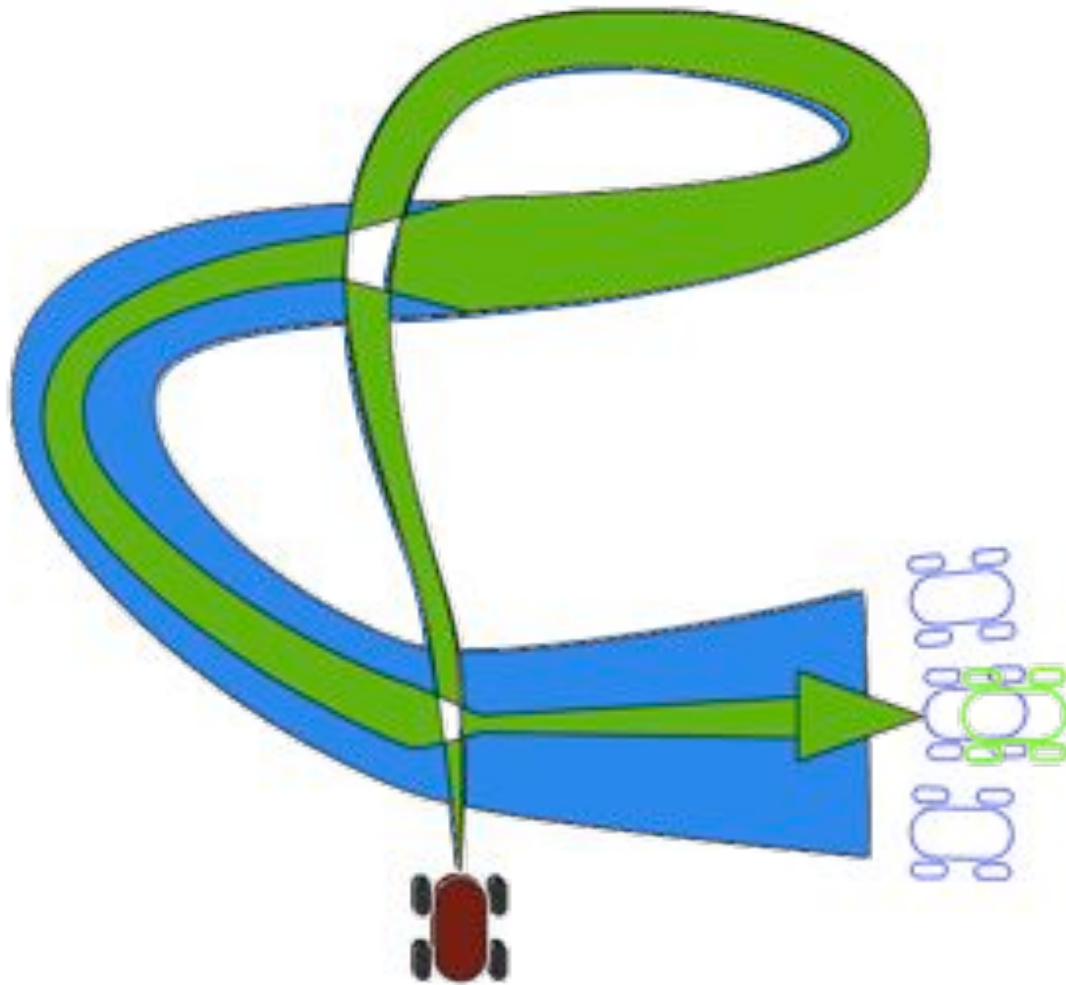
Principle of SLAM



- Landmark detection
- Relative observations (measures)
 - Of the landmark positions
 - Of the robot motions
- Observation associations
- Refinement of the landmark and robot positions

All these functions can be achieved in numerous different ways

Principle of SLAM



Dead-reckoning

SLAM

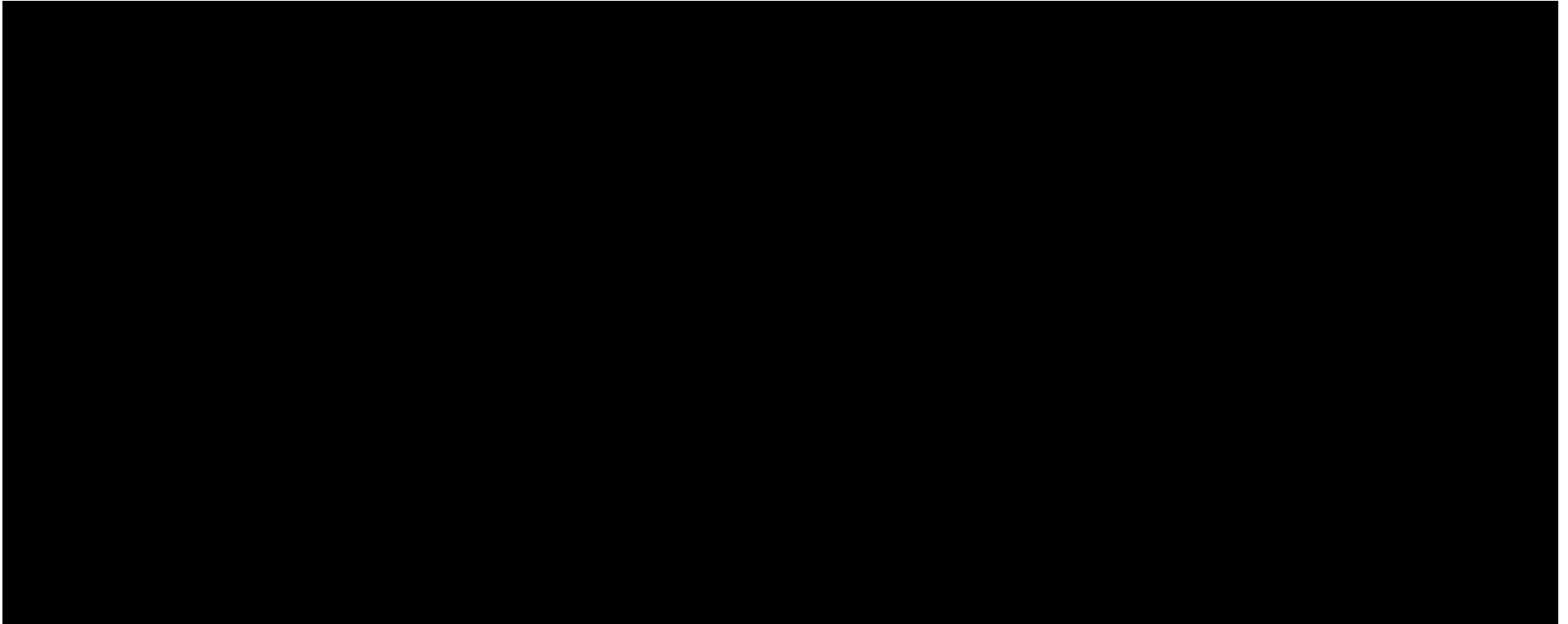
(a rich scientific history, with different communities involved)

SLAM comes in numerous flavors

Essential questions to answer:

- | | |
|--------------------------------|---|
| 1. Which sensors / landmarks? | Points / lines / planes / dense signal... |
| 2. Which estimation formalism? | EKF, IF, PF, Set membership, non linear optimization... |
| 3. In which frame? | Absolute / local / relative |

Illustration: vision-based SLAM



What is localization?

Essential questions to answer:

- | | |
|---|---------------------------------|
| 1. With which precision? | From <i>cm</i> to <i>meters</i> |
| 2. In which frame? | Absolute vs. local |
| 3. At which frequency? | From <i>kHz</i> to “sometimes” |
| 4. <i>Integrity</i> of the solution? | |
| 5. <i>Availability</i> of the solution? | |

cm accuracy,
> 100 *Hz*,
local frame

- Ensure the lowest level (locomotion) controls
- Ensure the proper execution of paths / trajectories
- Ensure the spatial consistency of the built models

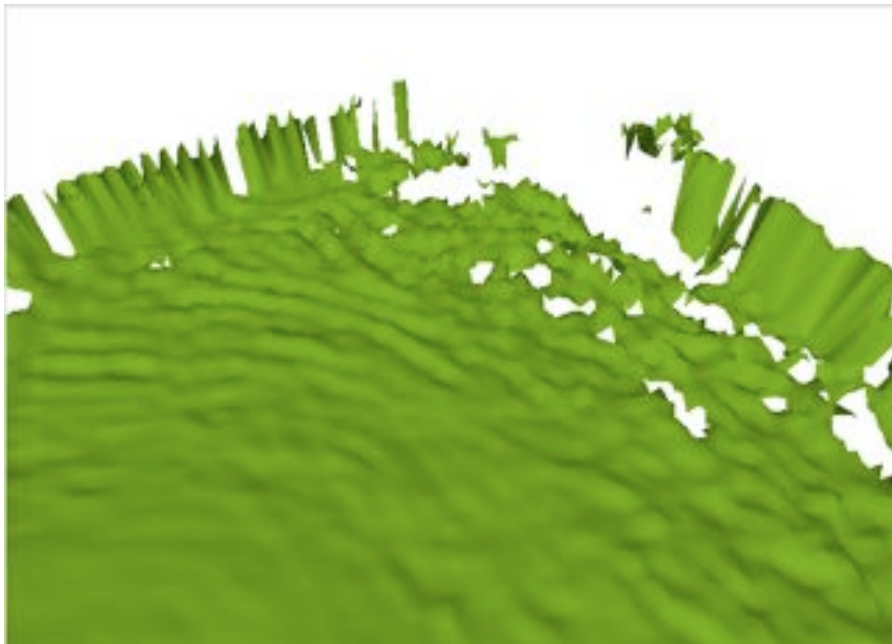
~*m* accuracy,
“sometimes”,
global frame

- Ensure the achievement of the navigation mission (“goto [goal]”)

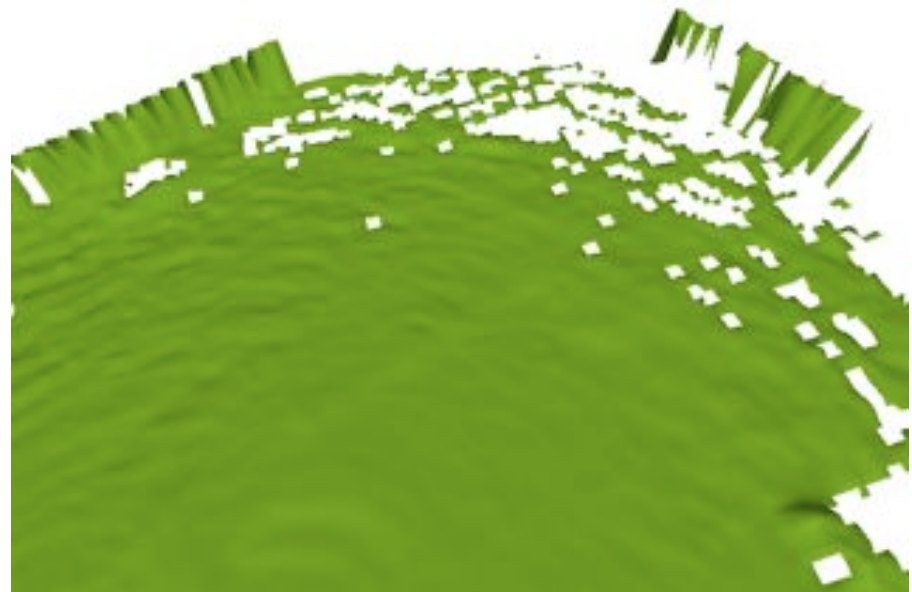
Localization precision required for a DTM

➔ DTM resolution $\sim 10\text{cm}$, height precision $\sim 3\text{cm}$

- Velodyne Lidar provides chunks of 64 points @ 3.5 kHz
- 1° error on pitch yields a 17cm elevation error @ 10m



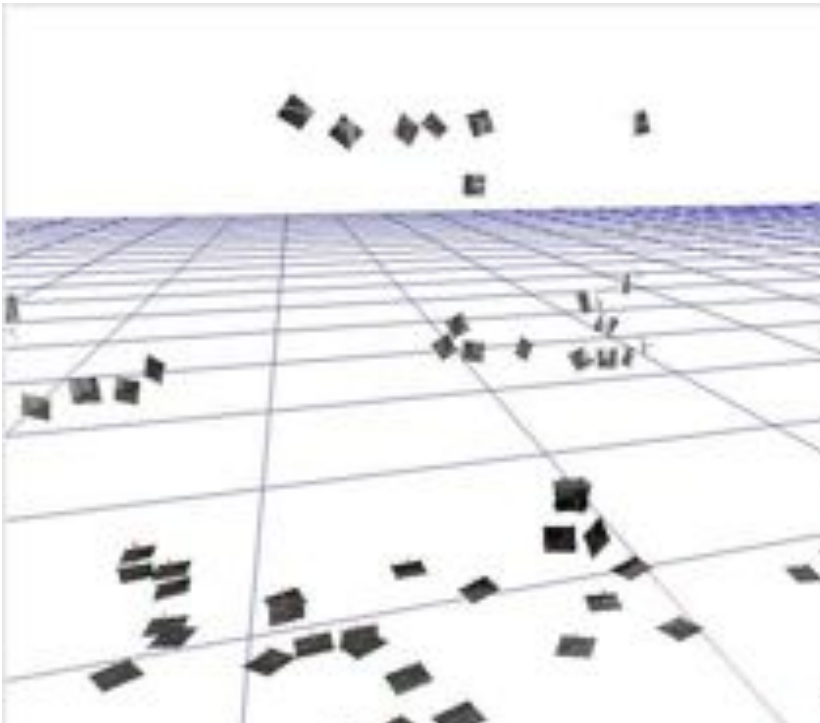
2m/s , GPS RTK @ 20Hz
+ Xsens AHRS @ 50Hz
+ FOG gyro @ 50Hz



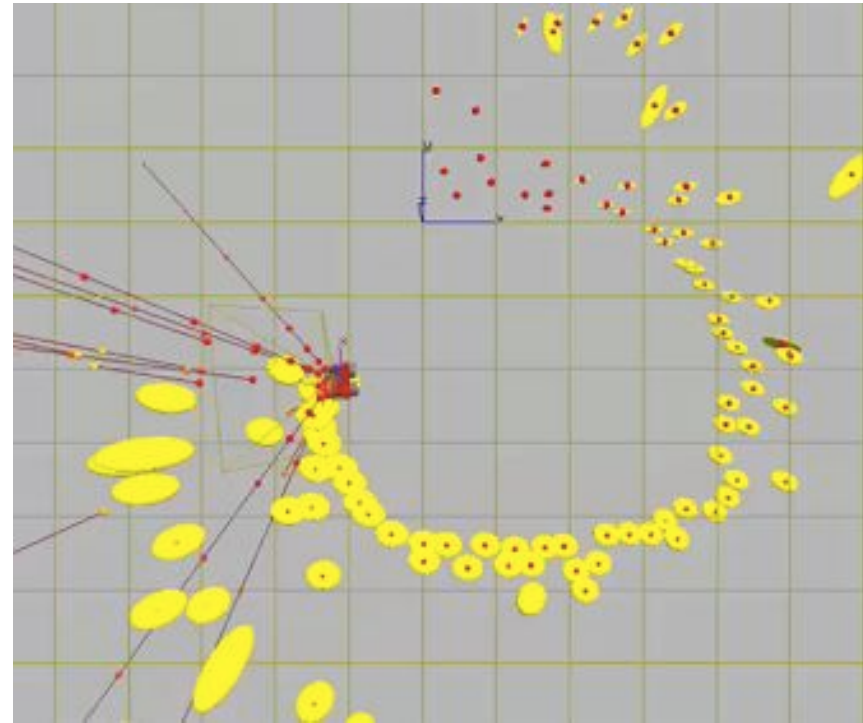
2m/s , RT-SLAM @ 100Hz

SLAM Maps

Twofold landmark representation:



1. Appearance



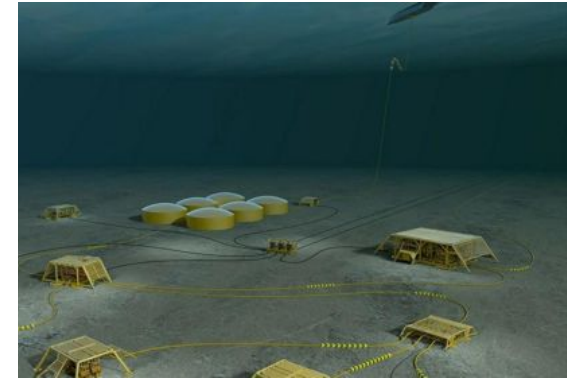
2. Positions and uncertainties

Outline

- Autonomous robots
- Mapping for autonomous navigation
- Mapping and localization
- **Beyond autonomous navigation**

Remote operations

- Sites / areas where human presence is precluded



Either not possible, too risky or too costly for humans

Remote operations

- Sites / areas where human presence is precluded
- To do what ?

Considered missions:

- exploration, search
- coverage / patrolling: observations, scene analyses, situation assessment
- *interventions* in the environment

In various application contexts:

- Environment monitoring (pollutions, science, ...)
- Search and rescue
- Civil security, defense applications
- ...

➔ *Information gathering and intervention* missions

Fleets of robots

- Advantages brought by robot teams
 - Increase of the achievable task and operation spaces
 - Higher robustness wrt. failures
 - Complementarities
 - Operational synergies
 - Robotic synergies

Fleets of robots

- Advantages brought by robot teams
 - Increase of the achievable task and operation spaces
 - Higher robustness wrt. failures
 - Complementarities
 - Operational synergies
 - Robotic synergies
- *E.g.* Air / Ground fleets of robots



“Remote eye” @ CMU



On going work @ ACFR



On going work @ LAAS & Onera

Challenges



Dozens of *heterogeneous* robots *cooperate* to achieve *long-lasting* missions in *large* environments

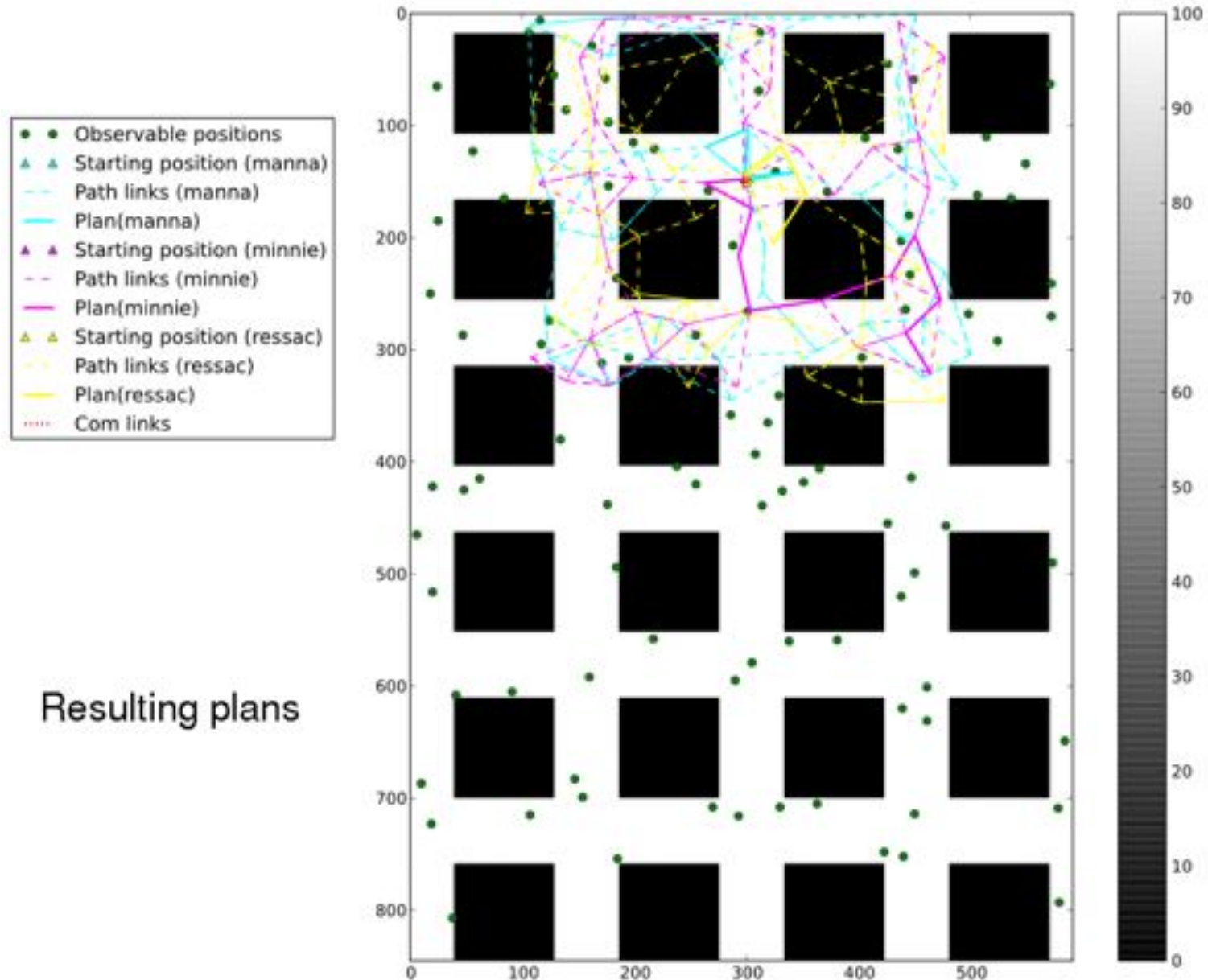
Large scale (km^3) implies:

- Faster robots, longer missions (“lifelong autonomy”)
- Large (multi-scale) environment models
- Communication constraints
- Cooperative and coordinated perception / planning / action

Illustration: planning a patrolling mission (1/2)



Illustration: planning a patrolling mission (2/2)



Which action models have been used?

- Robot motion model

$$\mathit{nav_time}(R_i, C_j, C_k) \in [0, +\infty]$$

$$\mathit{nav_cost}(R_i, C_j, C_k) \in [0, +\infty]$$

- Observation model

$$\mathit{obs1}(R_i, C_j) = \{C_1, \dots, C_n\} \subset M_{R_i}, C_j \in M_{R_i}$$

$$\mathit{obs_utility}(C_j) \in [0, 1], C_j \in M_{R_1} \cup \dots \cup M_{R_n}$$

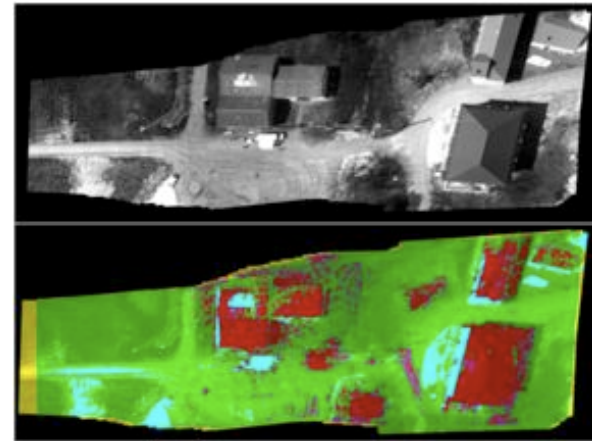
- Communication model

$$\mathit{com}(R_i, C_j) = \{C_1, \dots, C_n\} \subset M_{R_i}, C_j \in M_{R_i}$$

$$\mathit{com2D}(R_i, C_i, R_j, C_j) \in \{0, 1\}$$

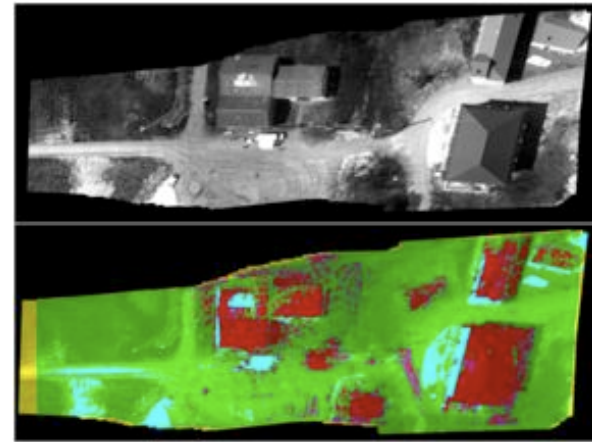
Which environment models have been used?

- To plan motions: express traversability / accessibility



Which environment models have been used?

- To plan motions: express traversability / accessibility



- To plan observations: line of sight visibility
- To plan communications: line of sight visibility

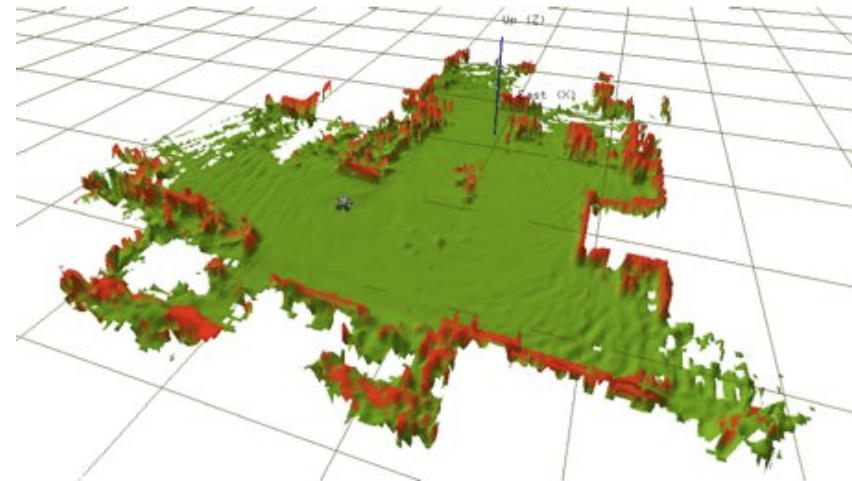
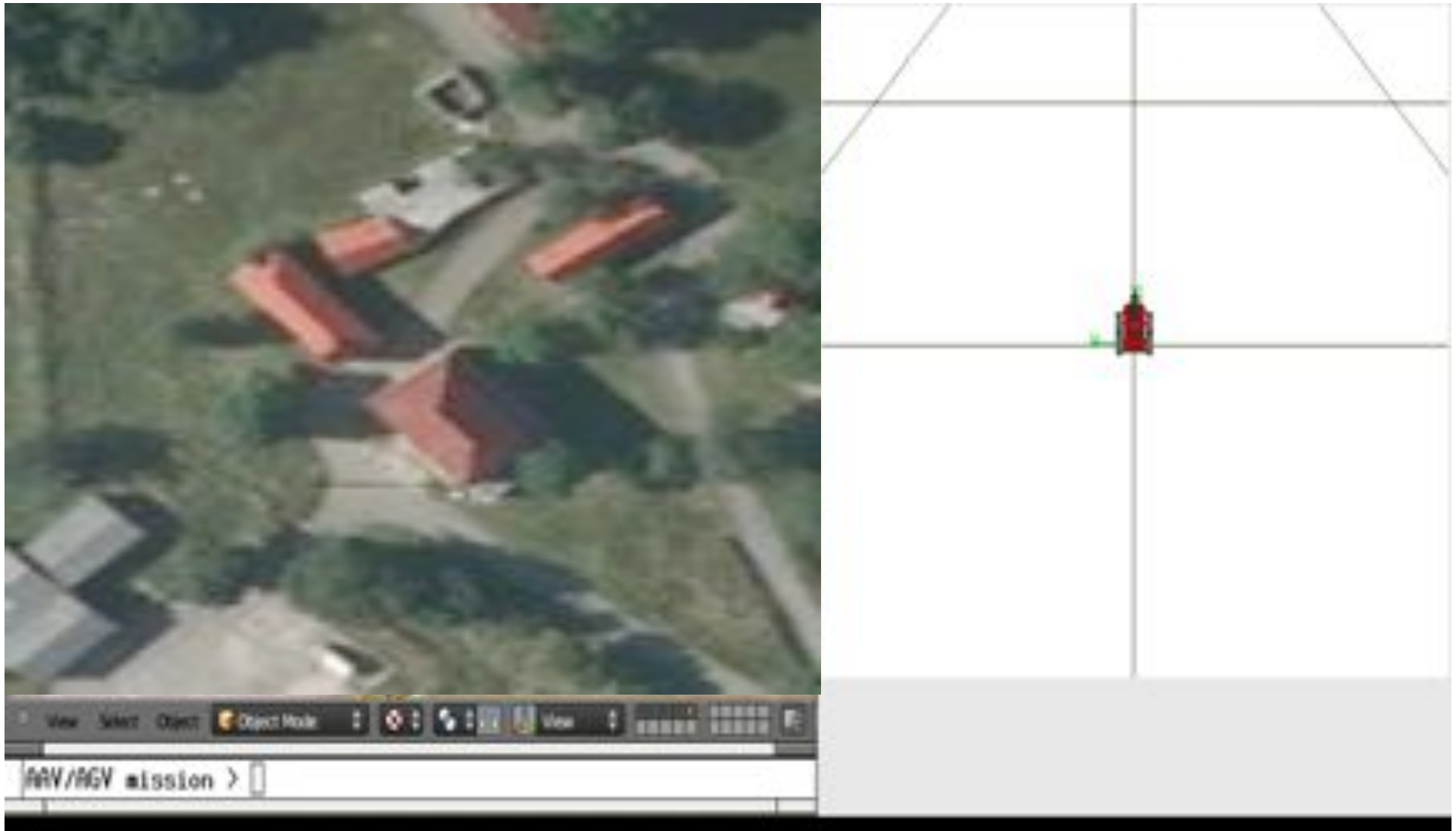
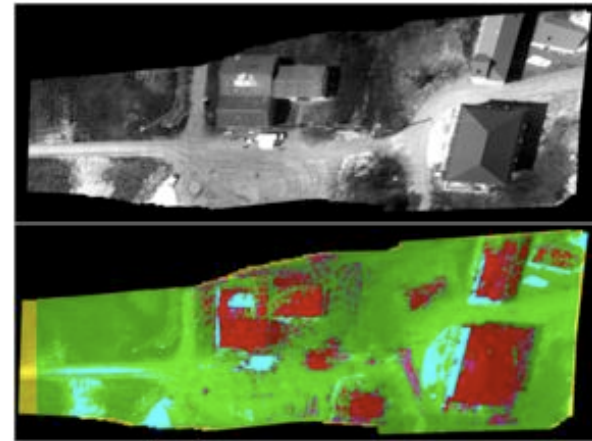


Illustration: autonomous navigation revisited



What environment models have been used?

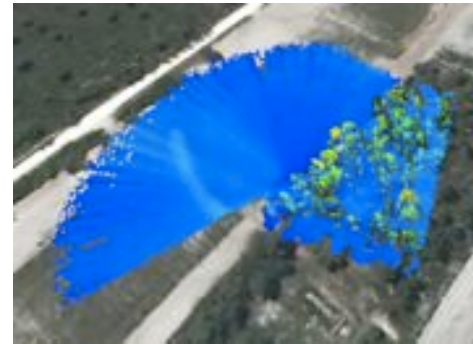
- To plan AGV motions: express traversability / accessibility



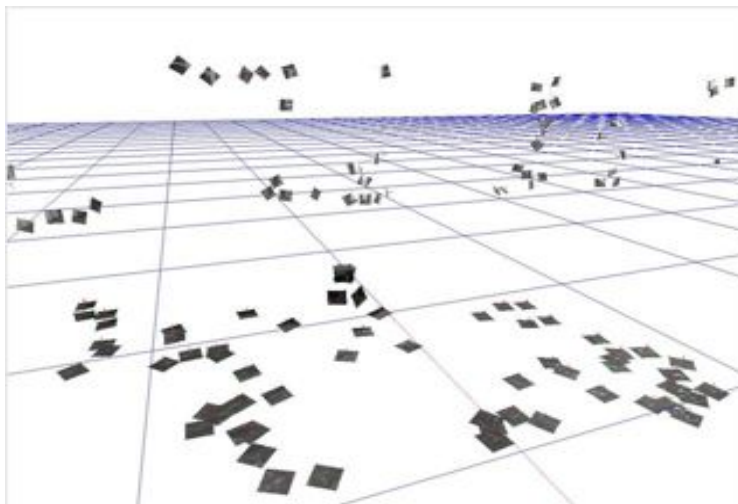
- To plan AUV motions: information quality (quantity) of the traversability model

Which environment models are used for localisation?

- Geometric models (a priori or built by the robots)

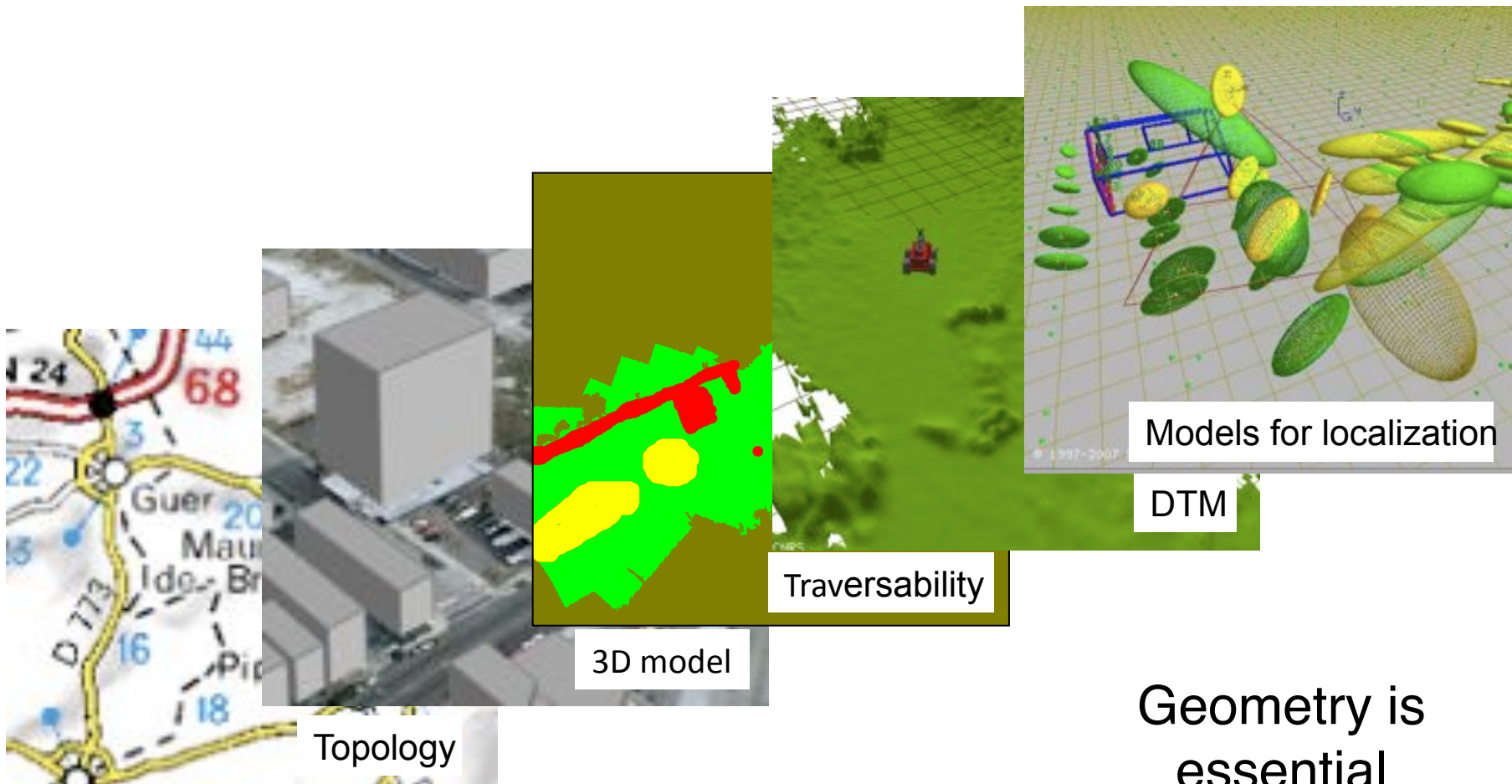


- Dedicated models (e.g. “landmarks”, “views memory”)



A database of environment models

- A variety of *dedicated* models have to be maintained



A database of environment models

- A variety of *dedicated* models have to be maintained
- A series of *queries* are exploited

Fonction	Action	Arguments	Returned value	Algorithm(s)
isAccessible	motion	r_m, p, c_{max}	$\{ p' \}$	Dijkstra
navigate	motion	r_m, p_1, p_2	$(path, c)$	A*, D*
canSee	perception	r_p, p	$\{ (p, \phi) \}$	ray tracing
test_comlink	com.	r_c, p_1, p_2	α	ray tracing

r_{action} action model

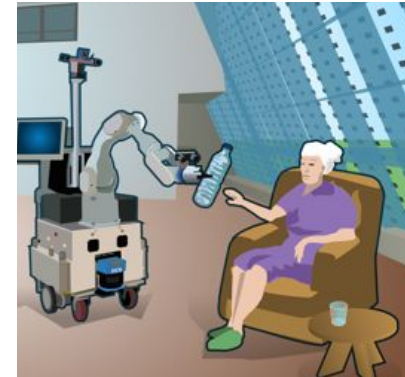
p position

c motion cost

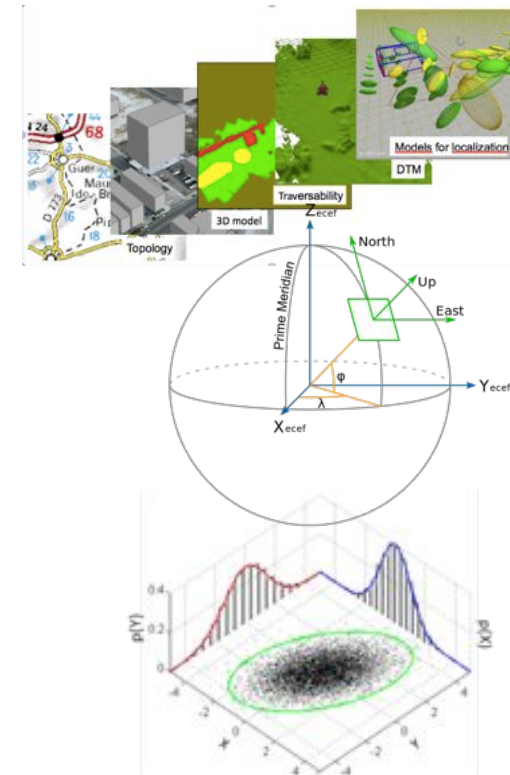
ϕ information quality

Summary

- Good environment representations are the foundations of good decisions – and hence of autonomy
 - Numerous tasks / missions can be turned into information gathering approaches
 - This also applies to personal robots and co-workers



- A *variety* of representations are required
- Geometry is certainly the most important information to represent (but not only)
- Representing the quality (precision, certainty) of information is essential



Current challenges

- The presence of operators will impose to share the information
→ spatial ontologies?
- Lifelong autonomy, lifelong mapping
- Richer semantics (from scene perception to scene interpretation)
- Exploit richer sensors (full-wave Lidars, hyperspectral imagery)
- Data flood
- Distributed real-time management of the spatial representations database