Autonomous teams of aerial and ground robots in search and rescue missions

Simon Lacroix
Labatory for Analysis and Architecture of Systems
CNRS, Toulouse
Context: teams of field robots
Air/ground robot teams

Usual advantages brought by robot teams
- Increase of the operation space
- Higher robustness wrt. Failures
- Complementarities
  ➞ Operational synergies
  ➞ Robotic synergies

UAVs assist UGVs
- Localization
- Communication relay
- Environment modeling
- ...

UGVs assist UAVs
- Detect clear landing areas
- Carry UAVs
- Provide energy support
- ...

“Remote eye” @ CMU
On going work @ ACFR
Mars2020 @ UPenn
Where and what for?

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

Considered missions:
- exploration, search
- coverage / patrolling: observations, scene analyses, situation assessments
- interventions in the environment

In various application contexts:
- Environment monitoring (pollutions, science, …)
- Search and rescue
- Civil security, defense applications
Where and what for?

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

Large scale \((km^3)\) implies:
- Faster robots, longer missions (“lifelong autonomy”)
- Communication constraints
- Large (multi-scale) environment models

Robot teams must not imply teams of operators!

➤ *A high level* of autonomy is required

(operators are not considered throughout this talk)
“On the importance of environment models”
Autonomous decision making in air/ground systems

Environment models

And yes, localization

(Mostly on-going work, with some unstable choices / ideas)
An elementary decision: AGV obstacle avoidance

Simple instance of a perception / decision / action loop:

- Gathering data on the environment
- Structuring the data into a *model*
- *Planning* the trajectory to find the “optimal” one
- Executing the trajectory
Simple instance of a perception / decision / action loop:

- Gathering data on the environment
- Structuring the data into a model

An elementary decision: AGV obstacle avoidance

Perception

Depth image  Digital terrain map
An elementary decision: AGV obstacle avoidance

Simple instance of a perception / decision / action loop:

- Gathering data on the environment
- Structuring the data into a **model**

**Perception**

- Planning the trajectory to find the “optimal” one

**Decision**

- Executing the trajectory

**Action**
An elementary decision: AGV obstacle avoidance

Simple instance of a perception / decision / action loop:

- Gathering data on the environment
- Structuring the data into a *model*

**Planning** the trajectory to find the “optimal” one

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

- Executing the trajectory
1. Planning a surveillance mission

Given:
- A team of robots
- An environment to monitor
- A set of constraints to satisfy (e.g. communications)

→ Find the (optimal) trajectories to observe the whole environment
1. Planning a surveillance mission

Given:
- A team of robots
- An environment to monitor
- A set of constraints to satisfy (e.g. communications)

Actions to plan:
- Observation tasks (hence motion tasks)
- Communications

Approach:
- A task allocation process (distributed market-based approach)
- Large scale: interleaving allocation and decomposition processes
1. Planning a surveillance mission
1. Planning a surveillance mission

Given:
- A team of robots
- An environment to monitor
- A set of constraints to satisfy (e.g. communications)

Actions to plan:
- Observation tasks (hence motion tasks)
- Communications

Required models:
- Of the observation tasks
- Of the robots motions
- Of the communications
2. Navigating a rover in an unknown environment

Given:
- A team of robots
- An unknown environment
- A set of constraints to satisfy \((e.g.\, \text{communications})\)

Find the (optimal) trajectory for the rover to reach a given goal
2. Navigating a rover in an unknown environment

Given:
- A team of robots
- An unknown environment
- A set of constraints to satisfy (e.g. communications)

Approach:
- The UAV serves the UGV, by providing traversability maps
- Find the areas to perceive by the UAV relevant for the mission

Actions to plan:
- Environment modelling tasks
- AGV and UAV Motions
- Communications
2. Navigating a rover in an unknown environment

Given:
- A team of robots
- An unknown environment
- A set of constraints to satisfy (e.g. communications)

Approach:
- The UAV serves the UGV, by providing *traversability maps*
- Find the areas to perceive by the UAV relevant for the mission

1. Run a A* search for the UGV
2. Integrate developed node costs
3. Evaluate the alternate paths, considering the UAV perception capacities
2. Navigating a rover in an unknown environment

Given:
- A team of robots
- An unknown environment
- A set of constraints to satisfy (e.g. communications)

Approach:
- The UAV serves the UGV, by providing *traversability maps*
- Find the areas to perceive by the UAV relevant for the mission

1. Run a A* search for the UGV
2. Integrate developed node costs
3. Evaluate the alternate paths, considering the UAV perception capacities
2. Navigating a rover in an unknown environment

(simulation with http://morse.openrobots.org)
2. Navigating a rover in an unknown environment

Given:
- A team of robots
- An unknown environment
- A set of constraints to satisfy (e.g. communications)

Actions to plan:
- Environment modelling tasks
- AGV and UAV Motions
- Communications

Required models:
- Of the traversability assessment function
- Of the robots motions
- Of the communications
3. Tracking a target in a known environment

Given:
• A team of robots
• A target locked by one robot (the “pursuer”)
• A known environment
• A set of constraints to satisfy (e.g. communications)

→ Find the (optimal) trajectories to keep the target in sight
3. Tracking a target in a known environment

Given:
- A team of robots
- A target locked by one robot (the “pursuer”)
- A known environment
- A set of constraints to satisfy (e.g. communications)

Actions to plan:
- Target “traps” (sentinel positions)
- Communications

Approach:
- The pursuer evaluate potential visibility losses

1. Target locked by the UAV
3. Tracking a target in a known environment

Given:
- A team of robots
- A target locked by one robot (the “pursuer”)
- A known environment
- A set of constraints to satisfy (e.g. communications)

Actions to plan:
- Target “traps” (sentinel positions)
- Communications

Approach:
- The pursuer evaluate potential visibility losses
Given:
- A team of robots
- A target locked by one robot (the “pursuer”)
- A known environment
- A set of constraints to satisfy (e.g. communications)

Actions to plan:
- Target “traps” (sentinel positions)
- Communications

Approach:
- The pursuer evaluate potential visibility losses

3. The UGV is asked for support
3. Tracking a target in a known environment

Given:

- A team of robots
- A target locked by one robot (the “pursuer”)
- A known environment
- A set of constraints to satisfy (e.g. communications)

Actions to plan:

- Target “traps” (sentinel positions)
- Communications

Approach:

- The pursuer evaluate potential visibility losses

4. Target locked by the UGV
3. Tracking a target in a known environment

Given:
• A team of robots
• A target locked by one robot (the “pursuer”)
• A known environment
• A set of constraints to satisfy (e.g. communications)

Actions to plan:
• Target “traps” (sentinel positions)
• Communications

Approach:
• The pursuer evaluate potential visibility losses
• Break the search complexity by exploiting redundancies in the tree

\[ O(m_t m_r)^h \text{ down to } O(h^3) \]
3. Tracking a target in a known environment

- Target (on the ground)
- Pursuer (UAV)
- Target positions where line of sight can be lost for a moment, with guaranteed recovery
- Target positions where line of sight can be lost without possible recovery
3. Tracking a target in a known environment

Given:
- A team of robots
- A target locked by one robot (the “pursuer”)
- A known environment
- A set of constraints to satisfy (e.g. communications)

Actions to plan:
- Target “traps” (sentinel positions)
- Communications

Required models:
- Of the robots and target motions
- Of the communications
Decision and environment models

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
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

<table>
<thead>
<tr>
<th>Surveillance</th>
<th>Rover navigation</th>
<th>Target tracking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Environment observations</td>
<td>Environment modeling</td>
<td>Target observations</td>
</tr>
<tr>
<td>Motions</td>
<td>Motions</td>
<td>Motions</td>
</tr>
<tr>
<td>Communications</td>
<td>Communications</td>
<td>Communications</td>
</tr>
</tbody>
</table>

Task allocation scheme | Heuristic graph search | Graph search + task allocation

Simulation = convolution of action and environment models

Environment models:
- at the heart of autonomy
- at the heart of cooperation
Autonomous decision making in air/ground systems
On the importance of environment representations
Autonomous decision making in air/ground systems

On the importance of environment representations

Environment models
Planning = Simulation + Search
• Simulation of the effects of an action with a predictive model
  ➔ by “convolving” action models with environment models

What are the main actions to plan / decide?

From an operations point of view:
• Motions
• Environment observations (payload)
• Communications (within robots, with the control station)

Plus, from a robotics point of view:
• Localization
• Environment perception and modeling
Decision and environment models

Planning motions

• At a coarse level (itinerary)
  ➔ notion of traversability
  (geometry, terrain nature)

• At a fine level
  ➔ geometry, terrain nature
  (e.g. digital terrain map)

Planning observations

• Need to predict visibilities
  ➔ geometry (2.5D or 3D)
Decision and environment models

Planning motions

• At a coarse level (itinerary)
  ➔ notion of traversability
  (geometry, terrain nature)

• At a fine level
  ➔ geometry, terrain nature
  (e.g. digital terrain map)

Planning communications

• Need to predict radio visibilities
  ➔ geometry, physical properties
  (or rather, learnt experience)
Decision and environment models

Planning localization

• GPS or corrections coverage
• INS / Odometry: terrain nature
• Exteroceptive sensors: landmarks or other models (geometry, appearance models, …)
Planning localization

- GPS or corrections coverage
- INS / Odometry: terrain nature
- Exteroceptive sensors: landmarks or other models (geometry, appearance models, …)

Planning environment perception & modeling

- Need to predict the *information gain*
  - amount of information in the environment models (uncertainty, entropy…)
A database of environment models

The key information is geometry
Exhaustive environment description

Geometry
Semantics
Physical properties
Chemical properties
Temperature, humidity...

Exteroceptive sensor data
Images
Point clouds
Radar echoes...

Environment models
Initial models (GIS)

‘‘Engineering autonomous agents [...] requires a steady flow of information from sensors to high-level reasoning components’’ [F. Heintz, ‘‘DyKnow’’]
Building a digital terrain model

With a rover, using point clouds

Resampling data to obtain a $z=f(x,y)$ representation on a regular Cartesian grid

(It is essential to maintain confidence / certainty / precision values during the process)

Using stereovision

Using a Velodyne lidar
Building a digital terrain model

With a UAV, using a Lidar

Resampling data to obtain a $z=f(x,y)$ representation on a regular Cartesian grid

[Paul Chavent @ Onera Toulouse]
Building a digital terrain model

With a UAV, using a camera

Up-to-date commercial bundle adjustment techniques
Building a traversability model

With a rover, using point clouds (here stereo)
Probabilistic labeling (Bayesian supervised learning)

- Possibility to introduce luminance and texture attributes
- Much more up-to-date classification or learning processes exist
Terrain models: data structures

“Raster” models: regular Cartesian grids

“Raster” models: hierarchical Cartesian grids or volumes

Graph structures easily derived
Terrain models: data structures

Triangular irregular meshes
Terrain models: 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
Merging air/ground models?

Traversability models

Digital terrain models

Inter-robot spatial consistency required
Autonomous decision making in air/ground systems
  On the importance of environment representations

Environment models
  On the importance of localization
Autonomous decision making in air/ground systems
  On the importance of environment representations

Environment models
  On the importance of localization

Localization
Localization is required to:

- Ensure the achievement of the missions, most often defined in localization terms ("goto [goal]", "explore / monitor [area]", ...)
- Ensure the lowest level (locomotion) controls
- Ensure the proper execution of paths / trajectories
- Ensure the spatial consistency of the built models
On the importance of localization
Localization solutions

A variety of available information:

• Motion sensors
  Odometry, IMU, velocimeters, …

• Environment sensors
  Lidar, camera(s), radar, …

• Infrastructure sensors
  GPS, radio receivers, …

• A priori information
  Motion models, environment models (maps), …
Localization solutions

A variety of available techniques:

• Dead-reckoning

• Map-based localization

• SLAM
But... what localization?

Essential questions to answer:

1. With which precision?  
   From cm to meters
2. In which frame?  
   Absolute vs. local
3. At which frequency?  
   From kHz to “sometimes”

- Ensure the lowest level (locomotion) controls
- Ensure the proper execution of paths / trajectories
- Ensure the spatial consistency of the built models
- Ensure the achievement of the missions, most often defined in localization terms ("goto [goal]", "explore / monitor [area]", ...)
But… what localization?

Essential questions to answer:

1. With which precision? From cm to meters
2. In which frame? Absolute vs. local
3. At which frequency? From kHz to “sometimes”
4. Integrity of the solution?
5. Disponibility of the solution?

- Ensure the lowest level (locomotion) controls
- Ensure the proper execution of paths / trajectories
- Ensure the spatial consistency of the built models
- Ensure the achievement of the missions, most often defined in localization terms (“goto [goal]”, “explore / monitor [area]”, …)

\[cm\] accuracy, @ > 100 Hz, local frame
\[\sim m\] accuracy, “sometimes”, global frame
Localization precision required for a DTM

- DTM resolution ~ 10cm, height precision ~ 3cm

- 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 @ 100Hz
+ FOG gyro @ 50Hz
Localization precision required for a DTM

- DTM built by an UAV with a Lidar

2m/s, GPS RTK @ 20Hz
+ INS @ x Hz
+ dynamic model
+ compass x Hz

During a calm day
Localization precision required for a DTM

- DTM built by an UAV with a Lidar

2m/s, GPS RTK @ 20Hz
+ INS @ x Hz
+ dynamic model
+ compass @ Hz

With a 10 km/h wind
Vision-based SLAM

http://rtslam.openrobots.org: a versatile EKF-based SLAM framework

1. Vision (monocular, stereoscopic, bi-cameras)
2. Point / line / planar landmarks
3. Predictions: motion model, INS
4. Additional observations: odometry (speed), GPS (position)
http://rtslam.openrobots.org: a versatile EKF-based SLAM framework

1. Vision (monocular, stereoscopic, bi-cameras)
2. Point / line / planar landmarks
3. Predictions: motion model, INS
4. Additional observations: odometry (speed), GPS (position)
http://rtslam.openrobots.org: a versatile EKF-based SLAM framework

- Real-time (100 Hz estimates, VGA @ 50Hz), active search
- Timestamp estimates through a dedicated filter
- IMU and calibration bias estimation
- Various landmark detection / observation / parameterization strategies
Localization precision required for a DTM

- DTM resolution $\sim 10cm$, height precision $\sim 3cm$

- Velodyne lidar provides chunks of 64 points @ 3.5 kHz:
  $1^\circ$ error on pitch yields a 17 cm elevation error @ 10m

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

2 m/s, RT-SLAM @ 100Hz
(known) SLAM issues

• SLAM processes complexity grows with the number of landmarks
  The map size can’t scale up

• The consistency of Kalman filter based solutions can’t be guaranteed
  The map size can’t scale up, loop closures may lead inconsistencies
Hierarchical SLAM [Tardos-2005], a graph of “submaps”:

- **Local** maps (EKF) of current vehicle pose and landmarks pose (nodes)
- **Global** map of relative transformations (edges)

**Local maps:**
- Fully correlated maps (robot and landmark states)
- No information shared between local maps
- Each map is initialized with no uncertainty
Hierarchical SLAM [Tardos-2005], a graph of “submaps”:

- **Local** maps (EKF) of current vehicle pose and landmarks pose (nodes)
- **Global** map of relative transformations (edges)

Global graph of maps:
- Robot’s pose
- The state is the **relative transformation** between local maps
- **Block diagonal** covariance before loop closure
Multi-robot multi-map hierarchical SLAM

Local level: A set of fully correlated submaps

Global level: A graph of map poses

Towards a distributed framework to integrate any localisation information
Multi-robot multi-map hierarchical SLAM

Various loop-closing events

“Rendez-vous”: inter-robot pose estimation

Absolute localization (GPS fix / localization wrt. an initial map)

Inter-robot landmark (or map) matches
SLAM issues

• SLAM processes complexity grows with the number of landmarks

  ➡️ The map size can’t scale up

• The convergence of Kalman filter based solutions can’t be guaranteed

  ➡️ The map size can’t scale up, loop closures may lead inconsistencies

• Detecting loop closures is an issue

  ➡️ Dedicated environment models are required
Detecting loop closures

Data association is mainly a perception problem

Powerful image indexing techniques (bag of words, e.g. FabMap)

Can be extended to Lidar scans (at least with global signatures)

Such robotcentric (or even sensorcentric) representations can not be shared / fused among robots

Landmark maps + image indices
Detecting loop closures between air/ground robots

Need to focus on the M of SLAM

→ **Geometry** is (again) the key
Points vs. lines in vision
Loop closures within air/ground robots

Inter-robot map matches
Loop closures within air/ground robots

“Rendez-vous”: inter-robot pose estimation
Keep the focus on geometric (3d, vectorized) representations
Keep the focus on geometric (3d, vectorized) representations

Integrate existing data (GIS)
Keep the focus on geometric (3d, vectorized) representations

Integrate existing data (GIS)

Distributed models
Management
  • APIs for clients
  • Maintain the inter-robot inter-model consistency
Keep the focus on geometric (3d, vectorized) representations

Integrate existing data (GIS)

Distributed models
Management
  • APIs for clients
  • Maintain the inter-robot inter-model consistency

Humans in the loop: information sharing (cf spatial ontologies)
Summary

Autonomous decision making in air/ground systems
  On the importance of environment representations

Environment models
  On the importance of localization

Localization
  On the importance of the environment representations
The following people have contributed to a small or large extent to the results and/or ideas discussed in this presentation:

*Rachid Alami, Guy Le Besnerais, Cyrille Berger, Sébastien Bosh, Redouane Boumghar, Hung Cao, Paul Chavent, Arnaud Degroote, Gilberto Echeverria, Malik Ghallab, Félix Ingrand, Thomas Lemaire, Naveed Muhammad, Bach Van Pham, Cyril Robin, Cyril Roussillon, Martial Sanfourche, Joan Solà, Teresa Vidal-Calleja*