Retour d’expérience sur une campagne d’acquisition de données

The Erfoud dataset

A comprehensive multi-camera and Lidar data collection for planetary exploration

LAAS CNRS

Simon Lacroix, Andrea De Maio, Quentin Labourey, Ellon Paiva Mendes, Pierre Narvor

magellium

Vincent Bissonette, Clément Bazerque, Fabrice Souvannavong, Raphaël Viards

esa

Martin Azkarate

Context

INFUSE

"Infusing data fusion in space robotics"

→ Tests planned at CNES premises

facilitators

Facilities for testing orbital and surface robotics BB

→ Tests planned on a planetary analogue

→ A LAAS / Magellium initiative : turn "tests at CNES" into "tests in Morocco", along with Facilitators

On the utility of datasets in robotics

- Data processing functions require a lot of data to be tested / validated / benchmarked
- Thorough datasets are hard to acquire
  → Datasets come without overhead for the developers
- Unbiased comparisons are more and more required
  → Datasets allow rigorous benchmarking
- Learning approaches are more and more favored
  → Datasets are needed for data-based learning

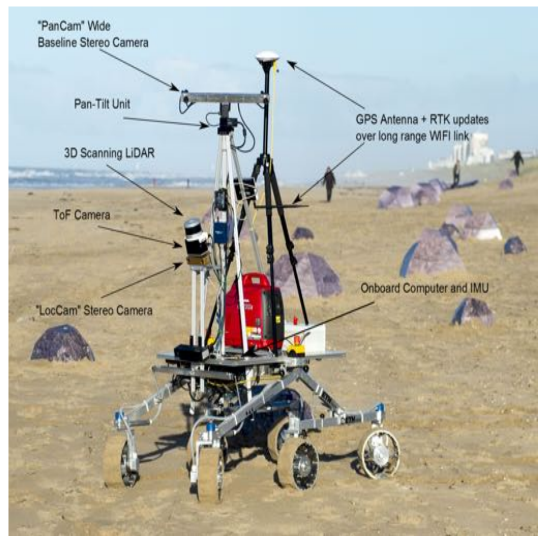
Yet an other robotics dataset?

- Dozens of high quality datasets are available

Yet an other robotics dataset?

- Dozens of high quality datasets are available
- But the offer for planetary robotics is restrained

Devon island
(Utoronto, 2012)



Katwijk Beach
(ESA, 2015)

Mt Etna
(DLR, 2017)

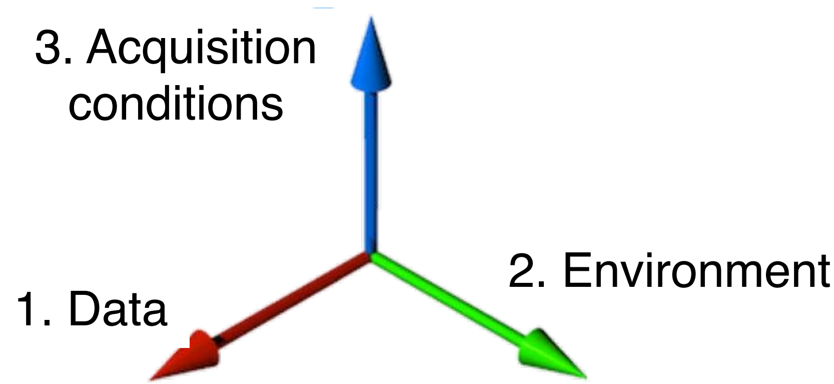
- NASA Planetary Data System

Good features for a robotics dataset?

- “Richness and quality”

Good features for a robotics dataset?

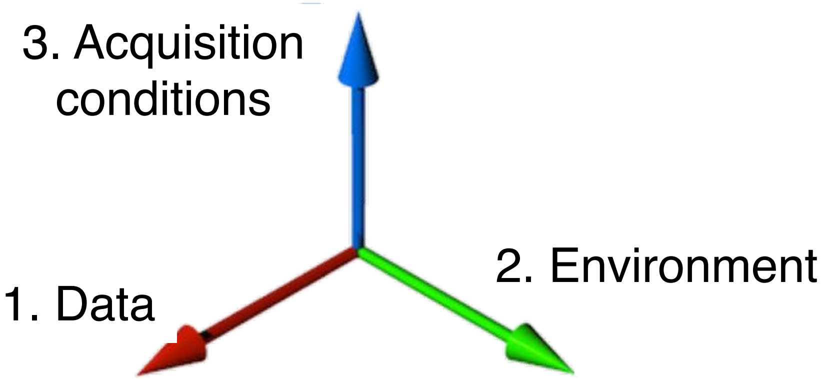
- Three essential dimensions



1. Data
    - Tailored to the targeted processes
    - Amount
    - Quality (*e.g.* timestamps, consistency)

Good features for a robotics dataset?

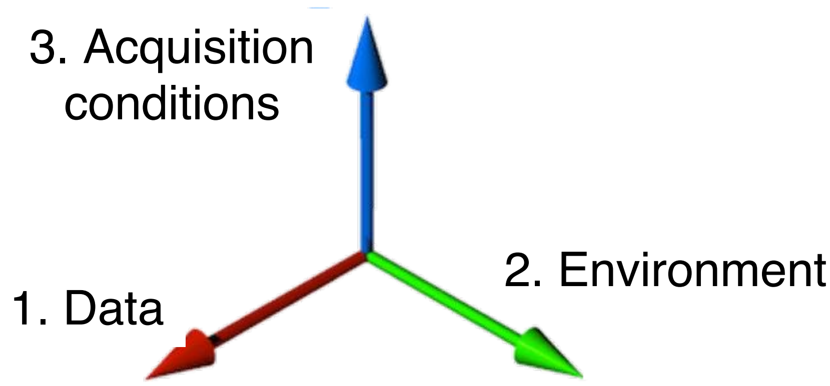
- Three essential dimensions



2. Environment
    - Representative of the targeted application context
    - Variety

Good features for a robotics dataset?

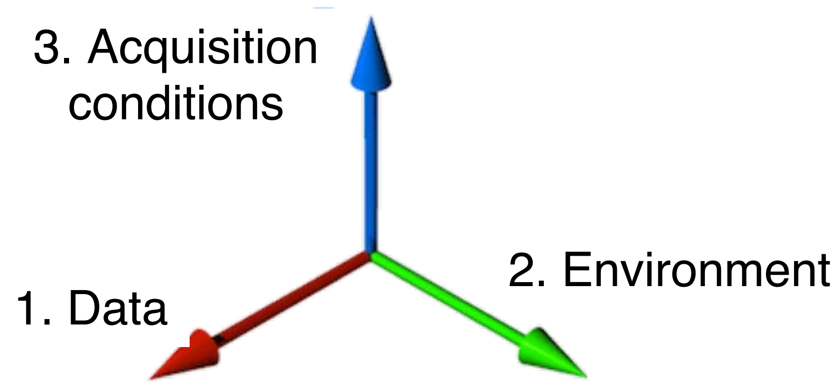
- Three essential dimensions



3. Acquisition conditions
    - Time of day
    - Data acquisition frequency, robot speed, type of trajectory…

Good features for a robotics dataset?

- Three essential dimensions



Two more dimensions:

4. Ground truths
5. Ease of use, ease of access

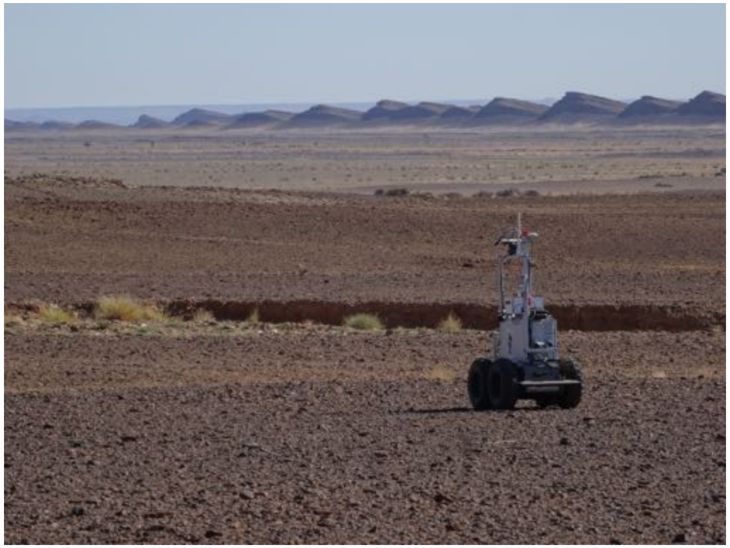
Outline

1. Environment
2. Data
3. Acquisition conditions
4. Ground truth
5. Usability and accessibility

1. The environment: near Erfoud, Morocco

1. The environment: three different sites

- Kesskess



1. The environment: three different sites

- Mummy

2. The environment: three different sites

- Merzouga

Outline

1. Environment
2. Data
3. Acquisition conditions
4. Ground truth
5. Usability and accessibility

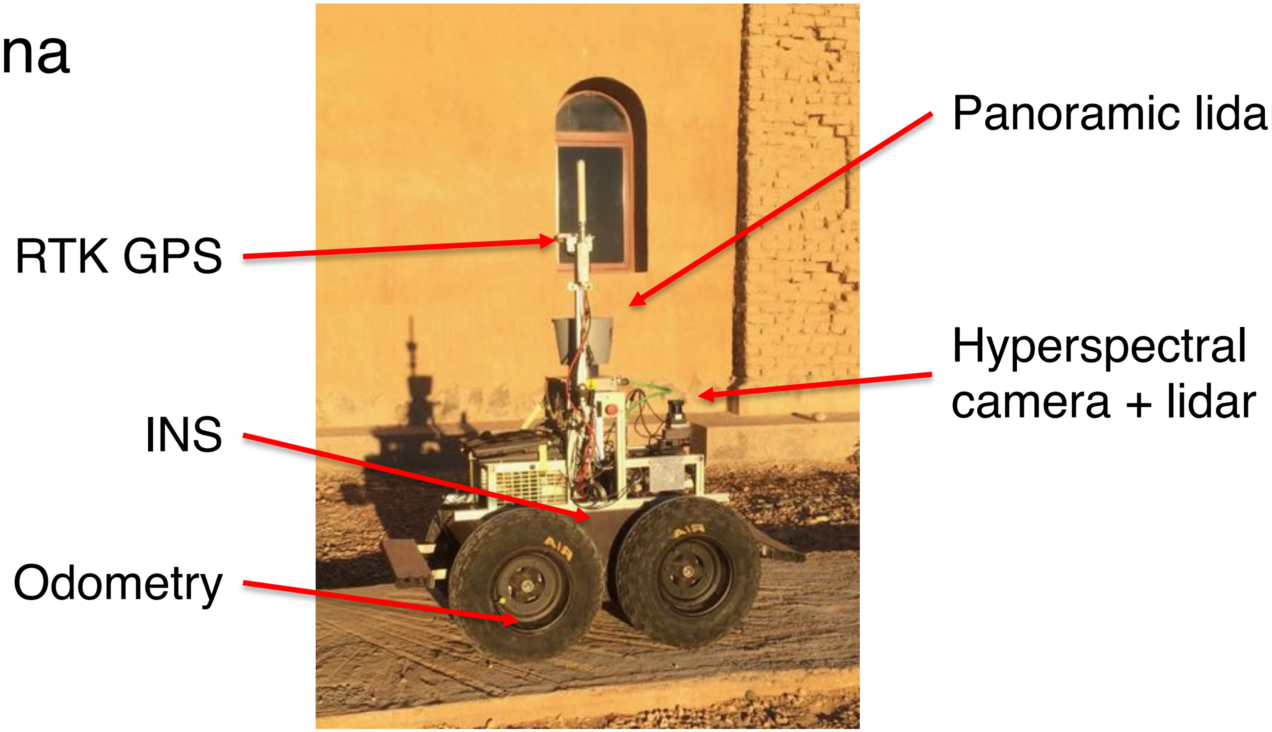
2. The data

- Targeted processes
  - Localization (in all its possible flavors: odometry, INS, visual odometry, SLAMs, absolute localization…)
  - Environment modelling processes (DTM, navigation map, …)
  - Declined in a multi-robot configuration

→ Use of two robots

2. The data sources

- The robot Mana



  - Segway RMP 400 platform (non steerable wheels without suspension)
  - i7 2.53 GHz CPU
  - Wifi
  - Radio-controlled emergency stop
  - Safety bumpers

2. The data sources

- The robot Minnie

  - Segway RMP 440 platform (non steerable wheels without suspension)
  - Two i7 2.53 GHz CPU
  - Wifi
  - Radio-controlled emergency stop
  - Safety bumpers

2. The data


navCam


Front hazCam


Rear hazCam

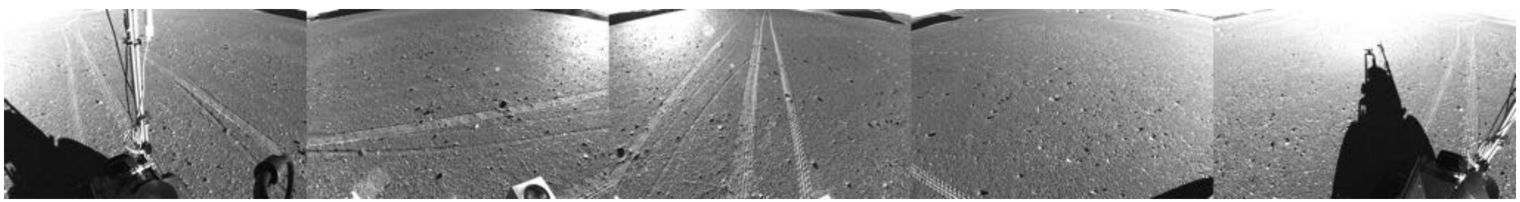
(all synchronized)

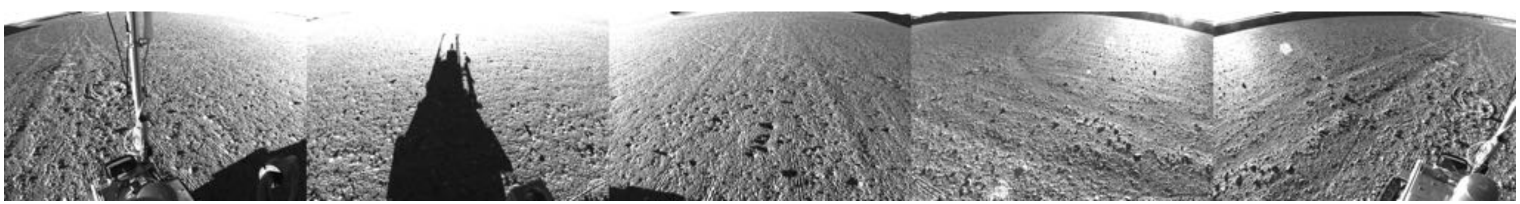
2. The data

navCam

Front hazCam

Rear hazCam





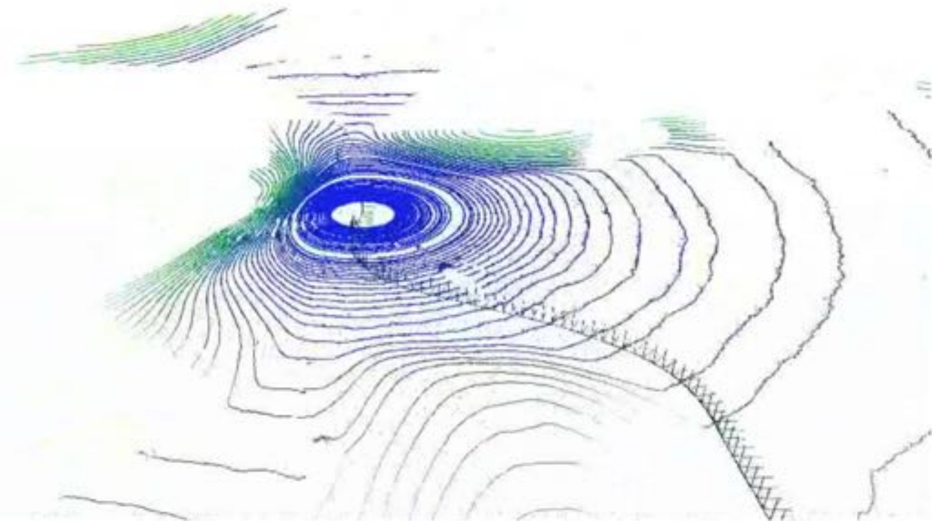
navCam panoramas

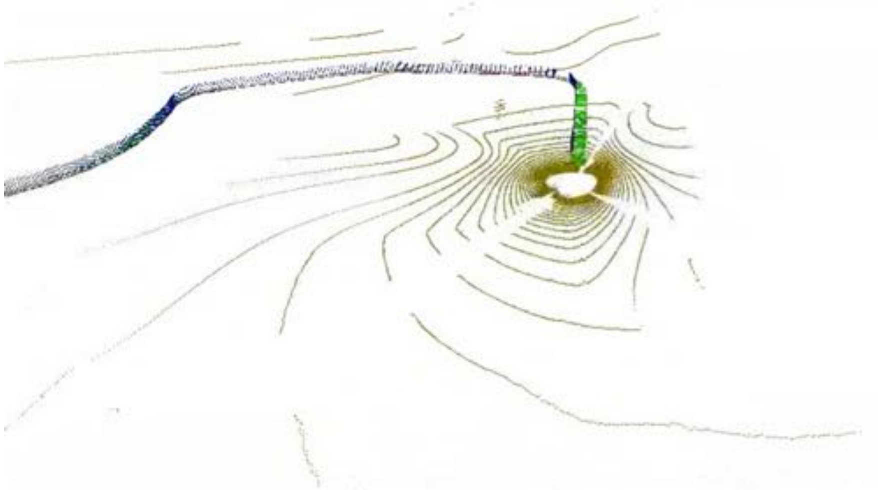
2. The data


navCam


Front hazCam

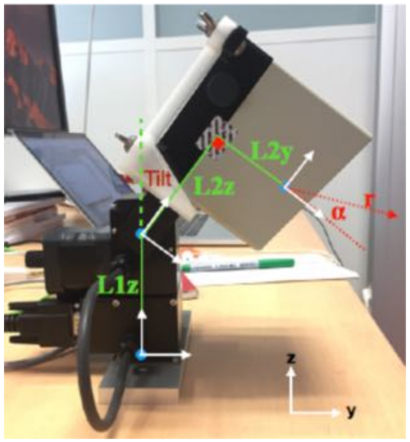

Rear hazCam

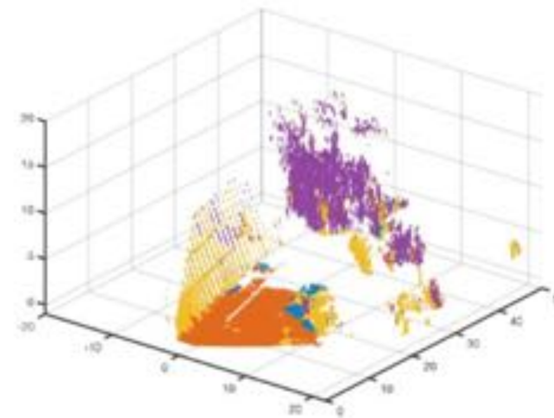

Velodyne HDL64 Lidar


Velodyne HDL32 Lidar

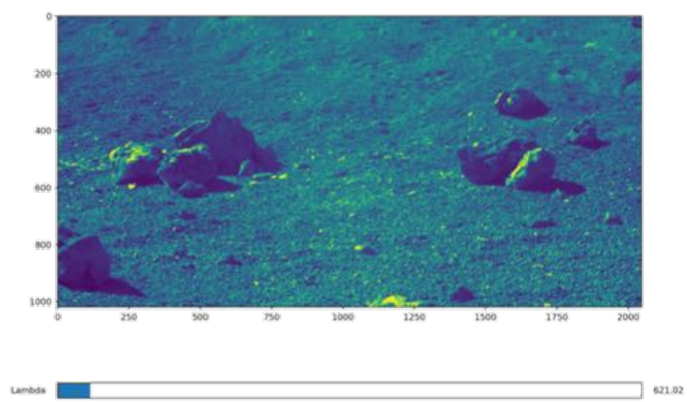
2. Unused sensors ☹

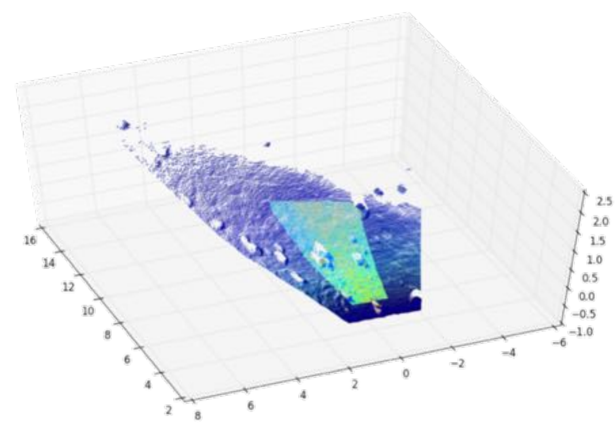
Dense long range Lidar





Hyperspectral camera + Lidar





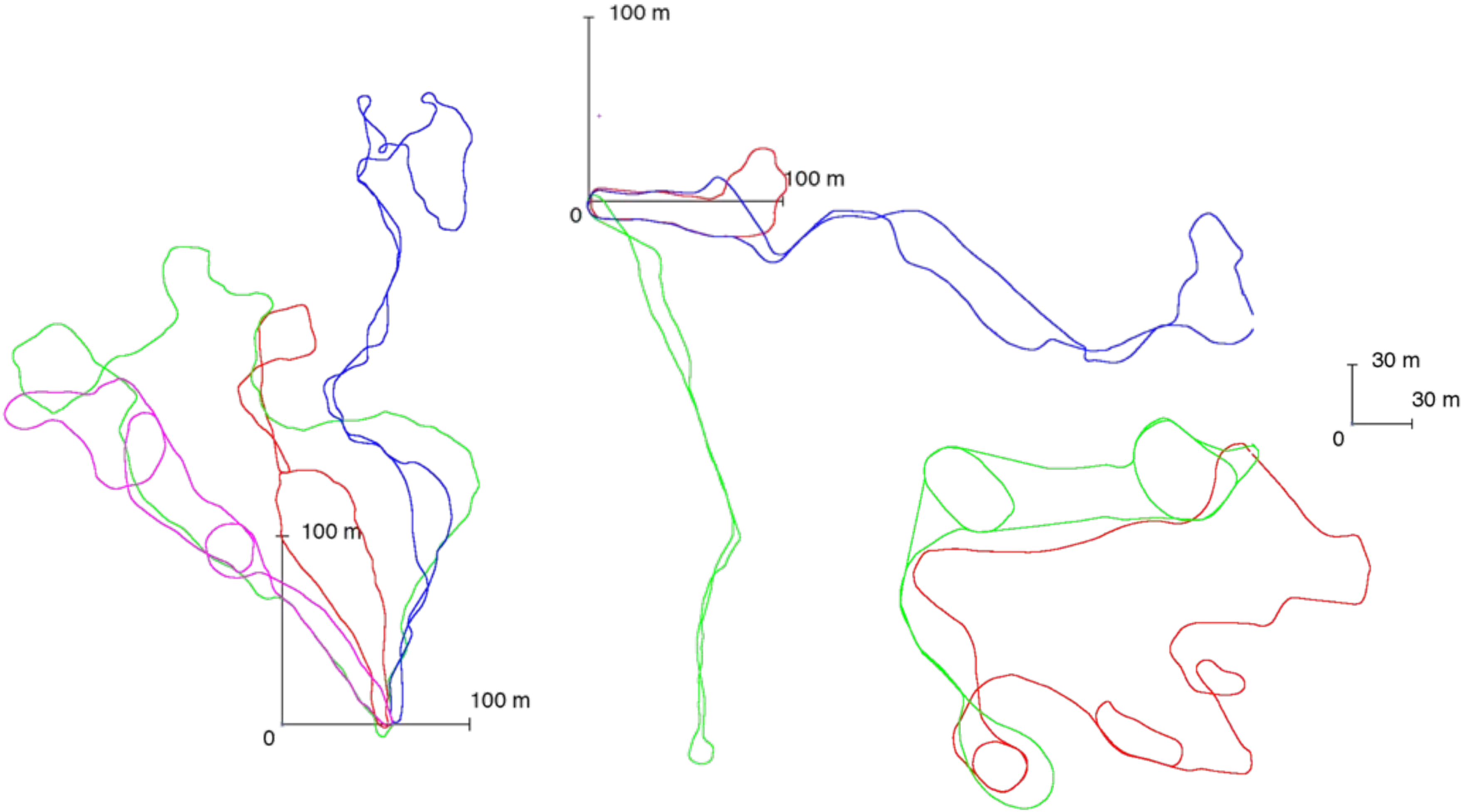
Outline

1. Environment
2. Data
3. Acquisition conditions

3. Acquisition conditions

3. Acquisition conditions

- “Record and replay trajectory” process
  - Various trajectories are defined (straight lines, loops, slopes…)

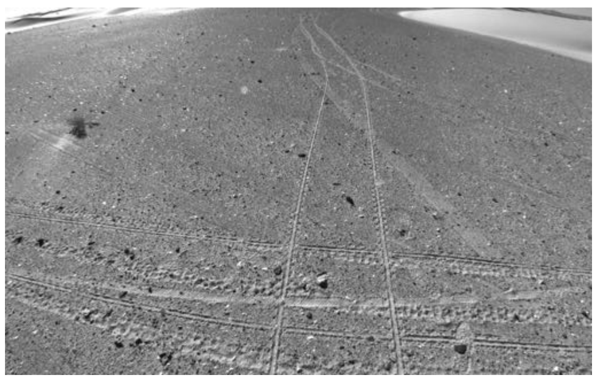


3. Acquisition conditions

- “Record and replay trajectory” process
  - Various trajectories are defined (straight lines, loops, slopes…)

  → Allows repeated data recordings (varying time of day, speed, sensor & robot)

Drawback: tracks on the ground

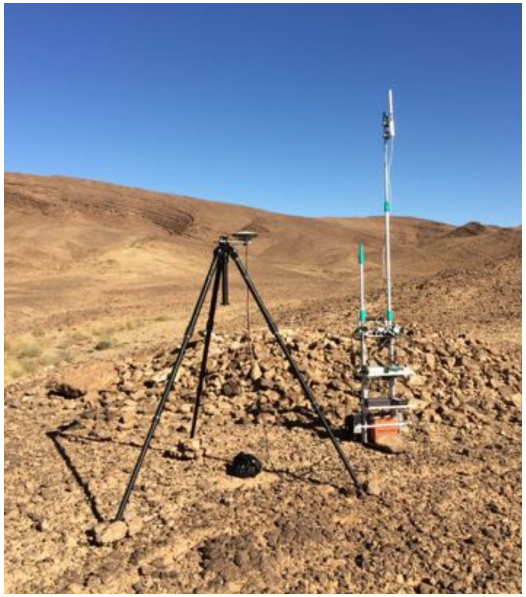


Outline

1. Environment
2. Data
3. Acquisition conditions
4. Ground truth

4. Ground truth

- Localization ground truth
  - Local terrain frame defined for each site
  - RTK GPS @ 20 Hz
  - Heading ground truth: FOG Gyro (drift of ~ 1°/hour)
  - Proper tagging of all acquired data



→ Absolute bias of ~ 0.5m for the local terrain frame

→ cm accuracy wrt. the local terrain frame

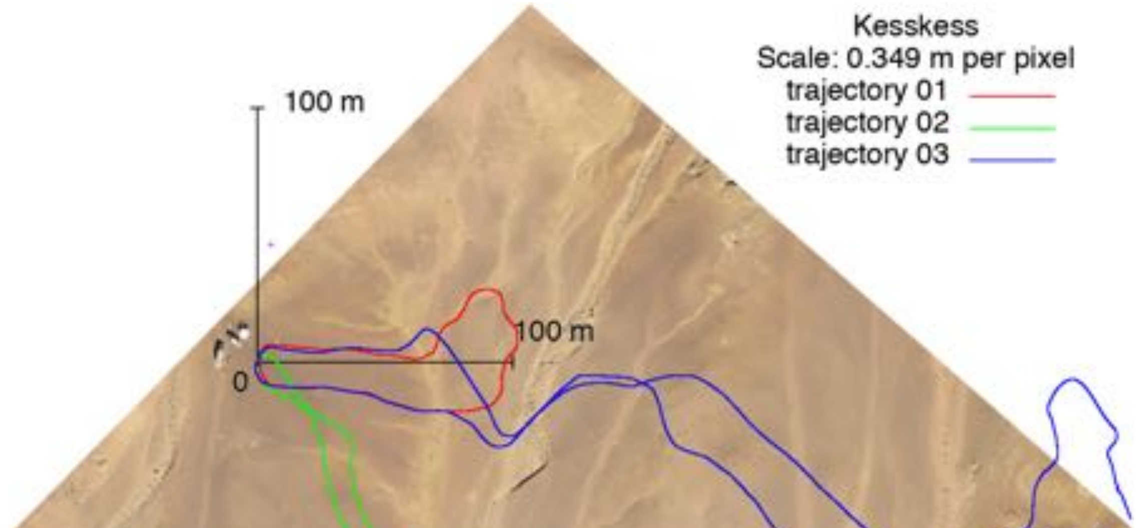
4. Ground truth

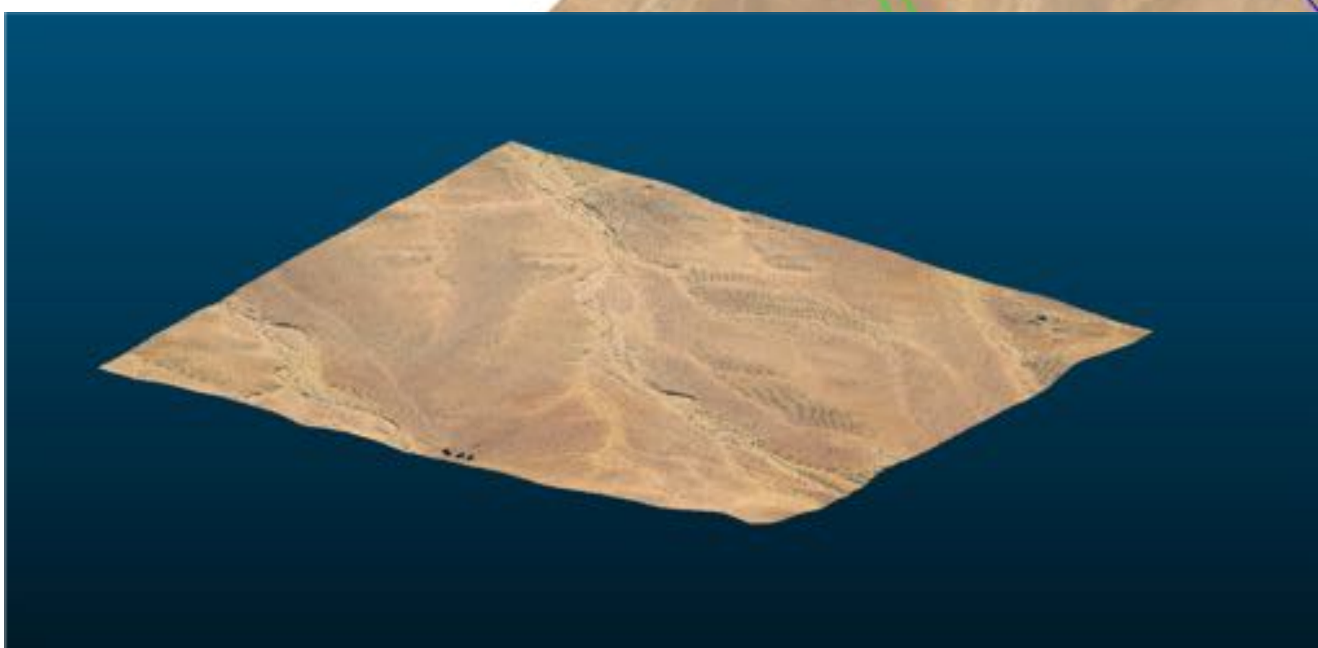
- Terrain ground truth
- SenseFly eBee UAV
  + PIX4D

→ 4 cm resolution DSM + orthoimage
(absolute localization bias of ~ 1.0 m)

4. Ground truth

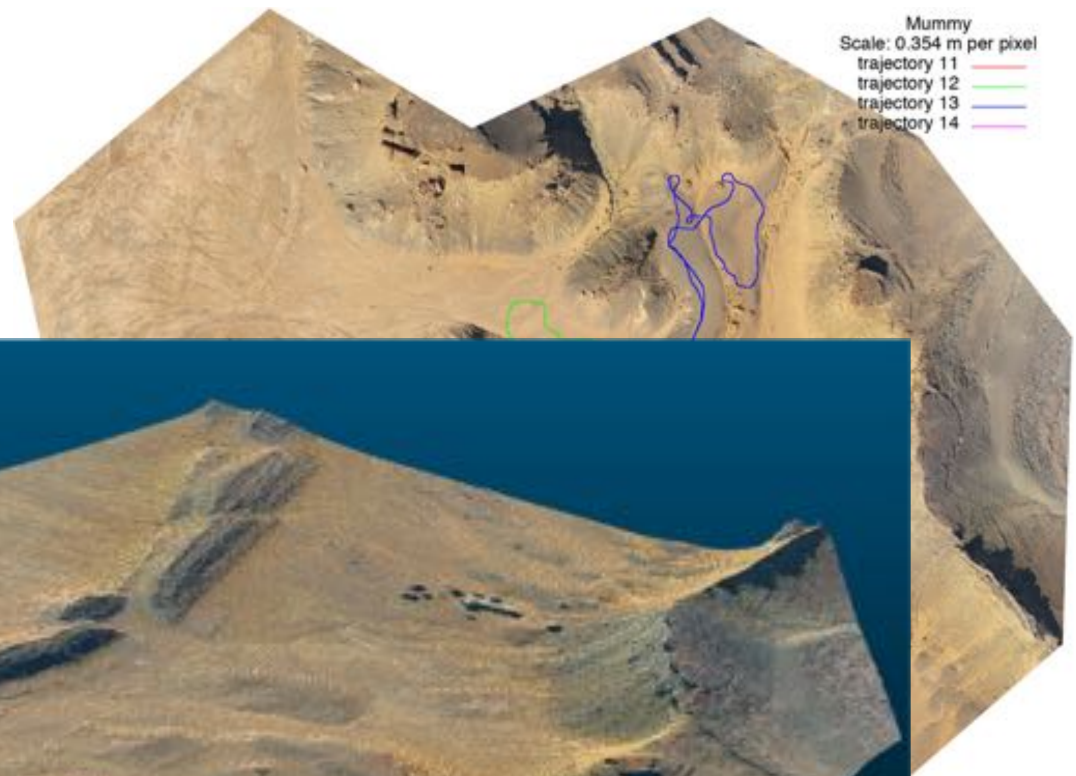
- Kesskess

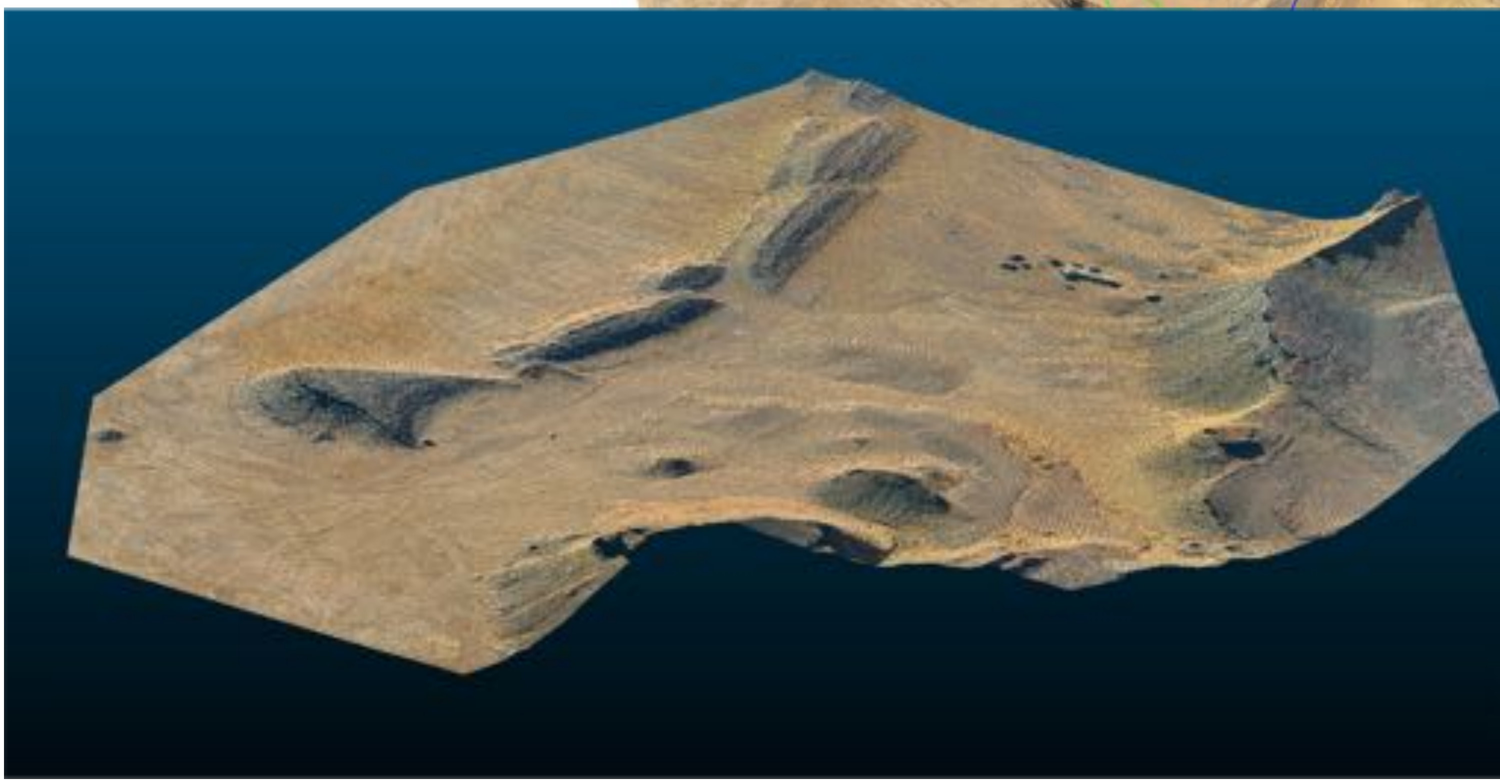




4. Ground truth

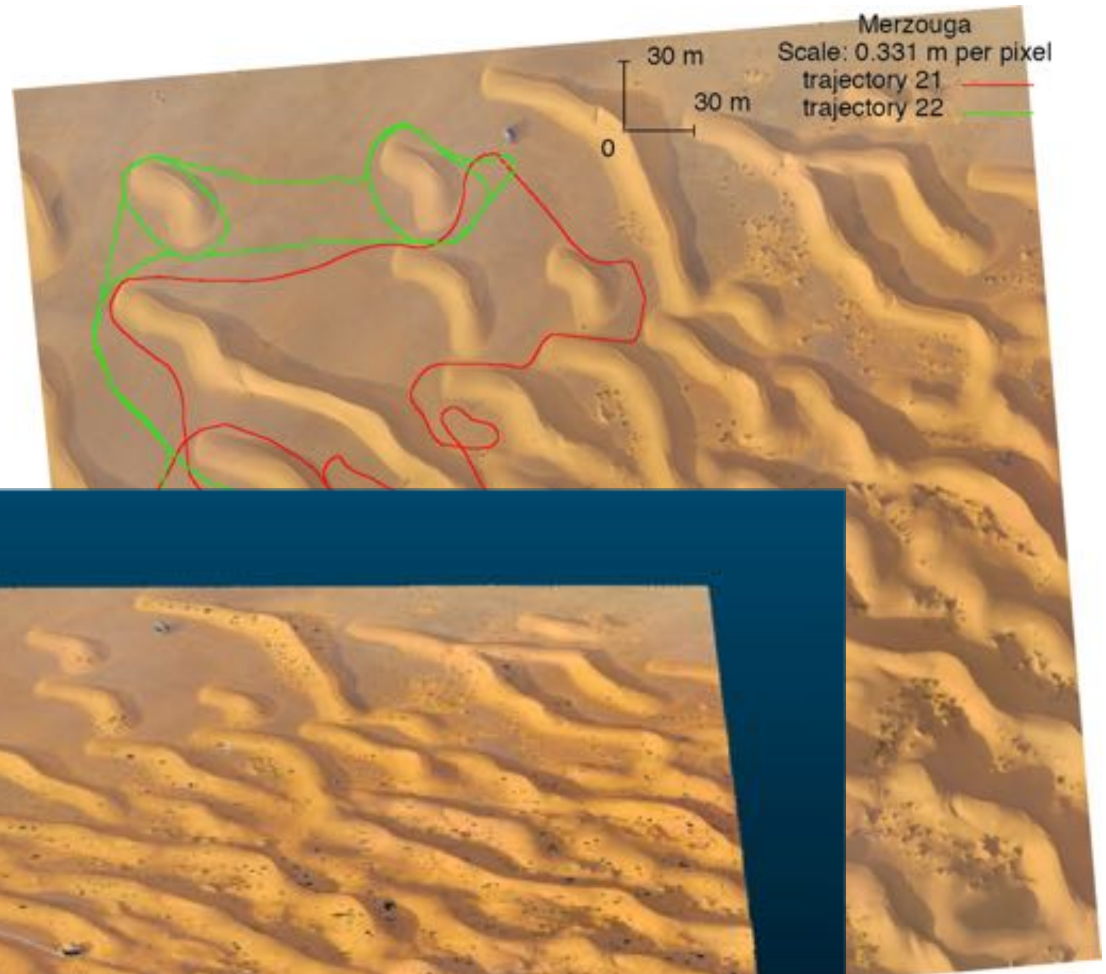
- Mummy





4. Ground truth

- Merzouga



5. Ease of use

- Data are provided as individual files
  - Raw images and .pcd files
  - Grouped into trajectory sequences
  - Always associated with metadata

```
#  1 - data_time_stamp
#  2 - data_utc_time
#  3 - robot_to_world_pose_time
#  4 - robot_to_world_pose_x
#  5 - robot_to_world_pose_y
#  6 - robot_to_world_pose_z
#  7 - robot_to_world_pose_qw
#  8 - robot_to_world_pose_qx
#  9 - robot_to_world_pose_qy
# 10 - robot_to_world_pose_qz
# 11 - robot_to_world_pose_roll
# 12 - robot_to_world_pose_pitch
# 13 - robot_to_world_pose_yaw
# 14 - robot_to_world_pose_sig_x
# 15 - robot_to_world_pose_sig_y
# 16 - robot_to_world_pose_sig_z
# 17 - robot_to_world_pose_curvilinear_abs
# 18 - robot_to_world_speed
# 19 - odometry_time
# 20 - odometry_x
# 21 - odometry_y
# 22 - odometry_z
# 23 - odometry_qw
# 24 - odometry_qx
# 25 - odometry_qy
# 26 - odometry_qz
# 27 - odometry_roll
# 28 - odometry_pitch
# 29 - odometry_yaw
# 30 - odometry_curvilinear_abs
# 31 - odometry_speed
# 32 - sensor_to_robot_pose_x
# 33 - sensor_to_robot_pose_y
# 34 - sensor_to_robot_pose_z
# 35 - sensor_to_robot_pose_qw
# 36 - sensor_to_robot_pose_qx
# 37 - sensor_to_robot_pose_qy
# 38 - sensor_to_robot_pose_qz
# 39 - sensor_to_robot_pose_roll
# 40 - sensor_to_robot_pose_pitch
# 41 - sensor_to_robot_pose_yaw
# 42 - cloud_number_of_points
# 43 - cloud_min_x
# 44 - cloud_max_x
# 45 - cloud_min_y
# 46 - cloud_max_y
# 47 - cloud_min_z
# 48 - cloud_max_z
```

5. Ease of use

- Data are provided as individual files
  - Raw images and .pcd files
  - Grouped into trajectory sequences
  - Always associated with metadata
- Calibration information (and data)
- Documentation, illustrations…

5. Ease of access

https://www.laas.fr/projects/erfoud-dataset/

- Considering alternate ways to access the data? GIS-like operations:
    - “Get all data that covers this area”
    - “Get all images heading westwards”
    - …

Summary of gathered data

- 9 trajectories on three sites (~ 7 km long)
- High resolution DSMs of the three sites
- 30 replays (aka datasets), overall distance ~ 20 km
- 90.000 lidar scans, 260.000 stereo pairs, 500 stereo panoramas (1 TB of raw data, ~ 3 TB all in all)

A glimpse at some data

- Things not expected on planets

A glimpse at some data

- Things not expected on planets

A glimpse at some data

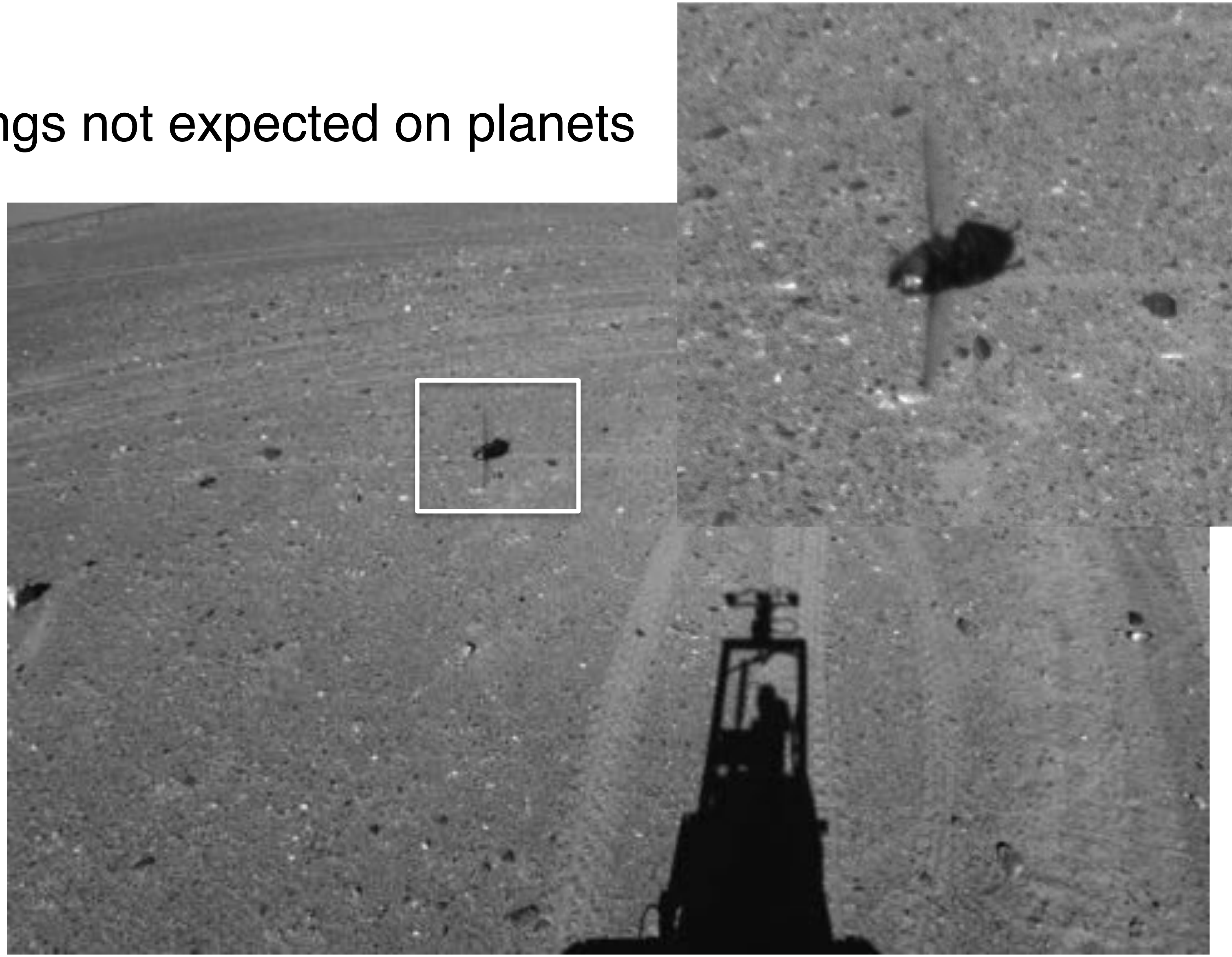
- Things not expected on planets

A glimpse at some data

- Things not expected on planets

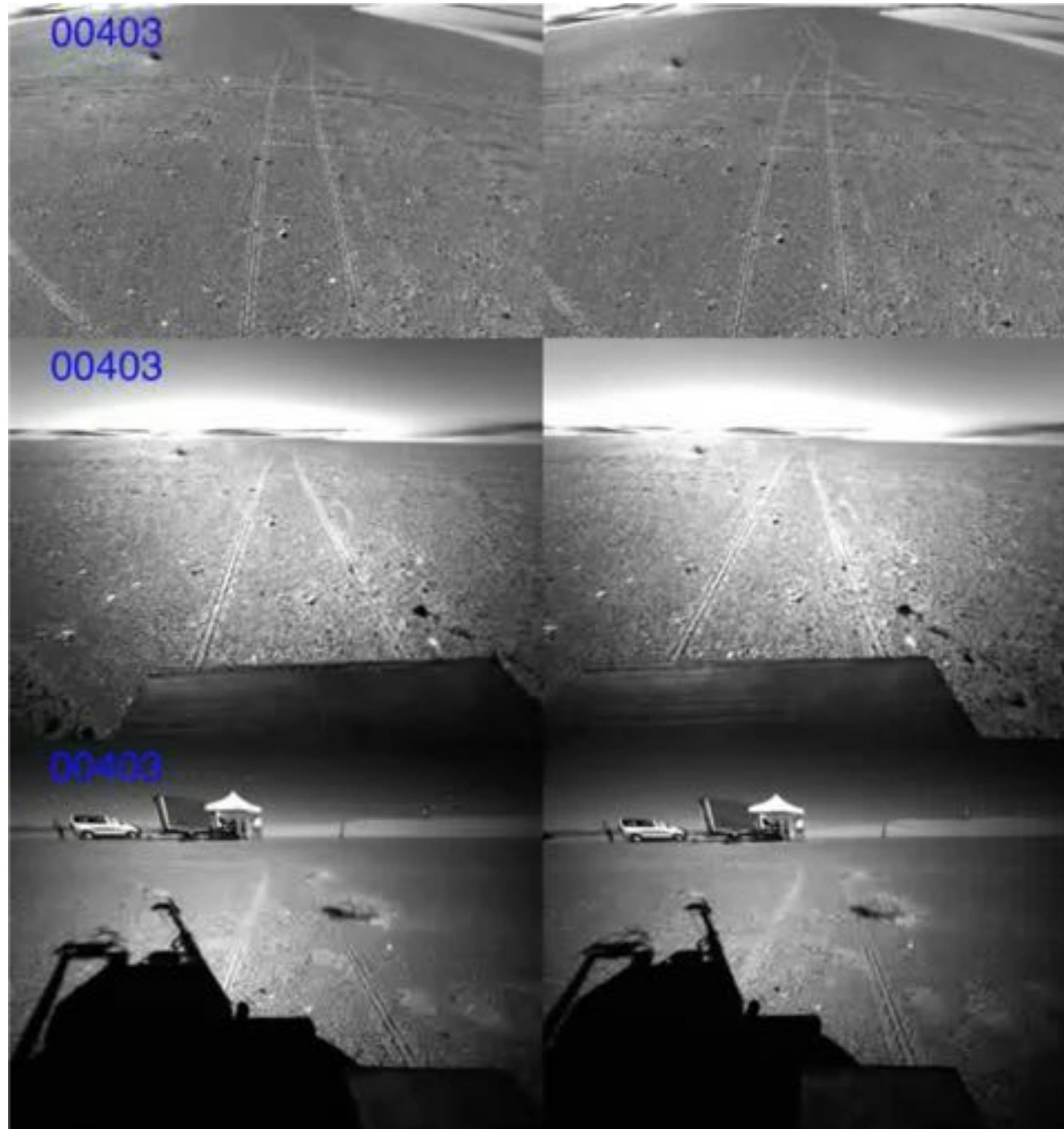
A glimpse at some data

- Things not expected on planets



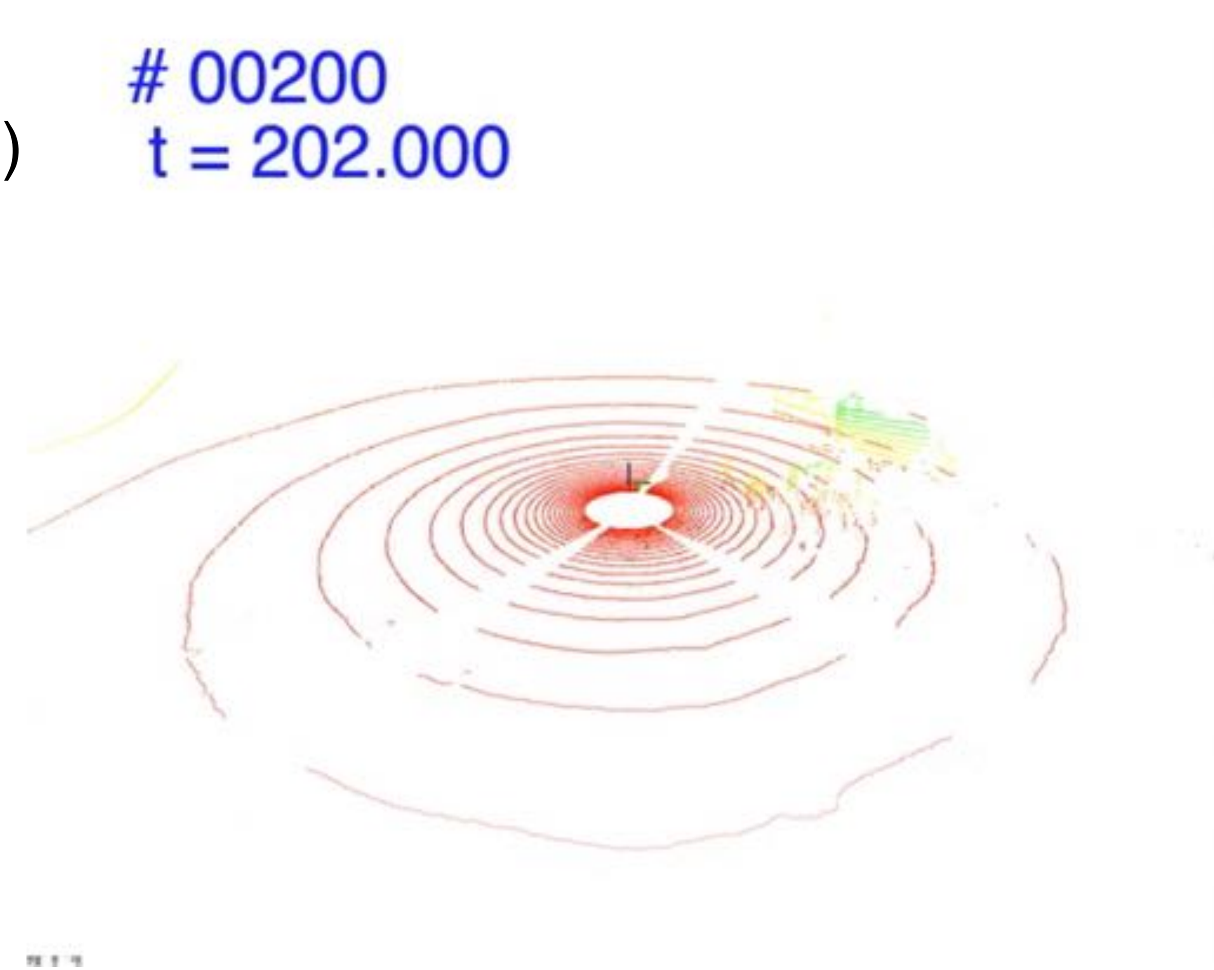
A glimpse at some data

- Bugs may turn into features (?)



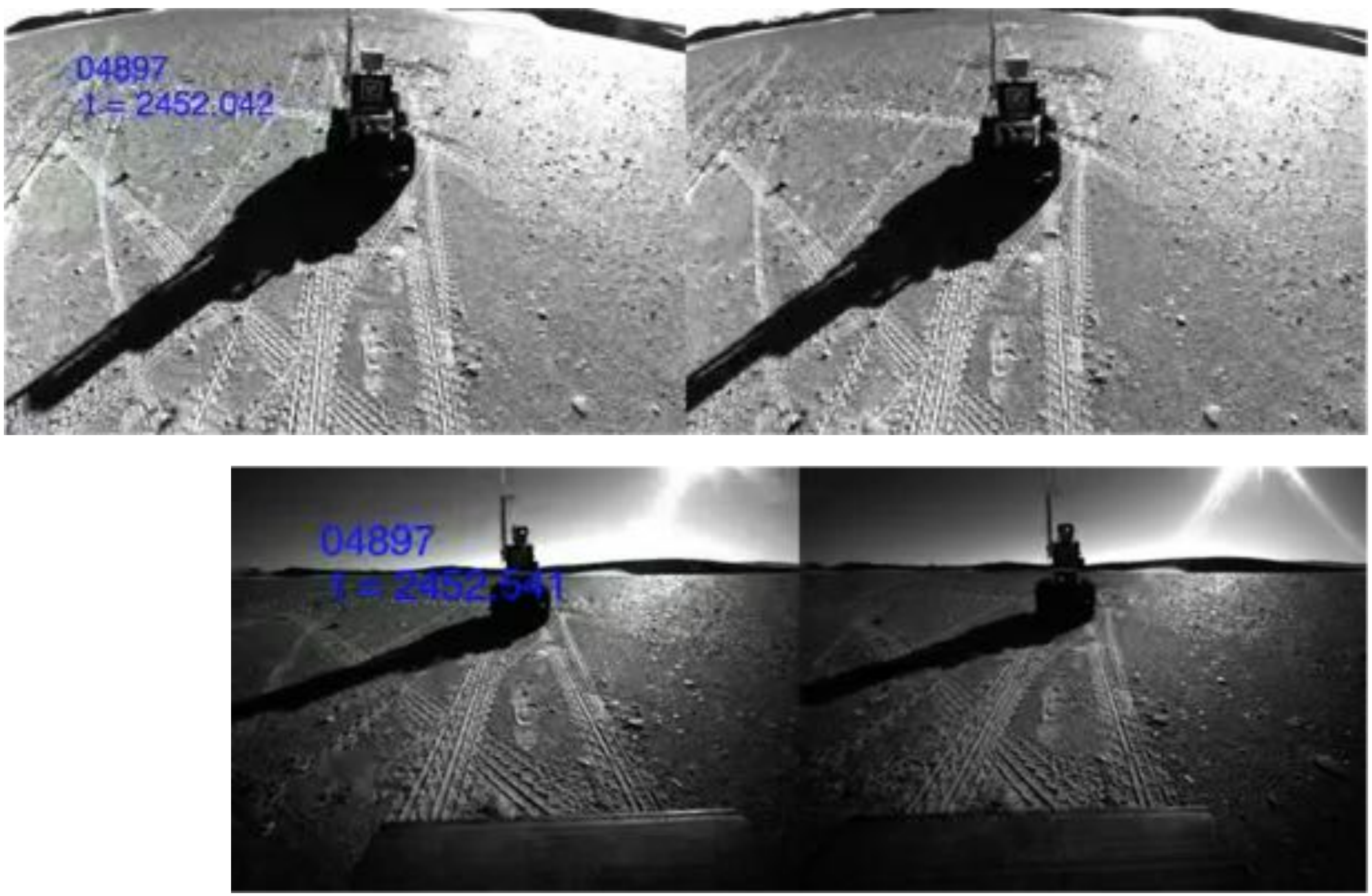
A glimpse at some data

- Bugs may turn into features (?)



A glimpse at some data

- Synchronous multi-robot data



Lessons learned

- As always: one should be able to deploy on the field without the source code…
- Always save (and provide) raw data – including calibration data
- Ease of use is key:
    - work hard on the documentation
    - Provide various means to get various data
- …