ARSI Deliverable D26.8

Autonomous navigation and data recording

Table of contents

1.1 References

[1] ARSI D26.1 — Operation requirements and system design
[2] ARSI D26.4 — Operational procedures and sewer inspection service
[3] ARSI D26.5 — Prototype for sewer inspection
[4] ARSI D26.6 — MAV prototype
[5] ARSI D26.7 — MAV platform verification for sewer inspections requirements
[6] PDTI Sewer — Final evaluation phase I
[7] 30 Hz video sample. Link: https://eurecatcloud-my.sharepoint.com/personal/pedro_cavestany_eurecat_org/_layouts/15/guestaccess.aspx?docid=161da5cfbeeb241c7aef1235ed459f55b&authkey=Aed82n0l5LbiLgTFAEcnZLw
[8] 5 Hz video sample. Link: https://eurecatcloud-my.sharepoint.com/personal/pedro_cavestany_eurecat_org/_layouts/15/guestaccess.aspx?docid=132e37cf9829f447daf5d8fa8df83ce7f&authkey=AYPiFv1OFqlbK0kNjdowGVU
[9] 3D reconstruction results. Link: https://youtu.be/1oTrckkjlHc
[10] Offline visualization app snapshots: https://invis.io/DKDT83H4J
[11] Offline visualization app behaviour: https://eurecatcloud-my.sharepoint.com/personal/toni_torrente_eurecat_org/_layouts/15/guestaccess.aspx?folderid=1265678f83614463b8429c89dee0b9e85&authkey=AdxwkGu_5rINfApojE5VKzM

1.2 Acronyms and abbreviations

- ARSI: Aerial Robot for Sewer Inspection
- MAV: Micro Aerial Vehicle
- RGB-D: Orbbec Astra RGB and depth camera
- ROS: Robotic Operating System
- RVIZ: ROS visualization interface
- RTABMAP: Real Time Appearance-Based Mapping

1. Introduction and scope

This deliverable describes the work done towards the usability of the platform developed, the data generated during the missions and its visualization. The result of this work is a functional prototype that enables a sewer operator to perform missions with the MAV and analyse the data generated.

The usability is addressed with the mission control interface, designed to handle the flight of the MAV. ARSI, the software that runs on-board, maintains a data link via wireless with the base station. It is therefore possible to send commands to ARSI from the base station, as well as receive data for monitoring and supervising the course of the flight. The mission control interface is the tool that will be handled by the user at the base station to view all the relevant data sent from ARSI and command the MAV.

The main extereoceptive sensor in the final set up in this phase is a VGA RGB-D camera, used to extract visual and depth information from the sewer. An HD resolution camera will be deployed in phase III and the camera configuration designed in phase I will be considered for deployment in phase III.

The visualization of the data generated during the missions is accomplished with an app that shows all the relevant data in a user-friendly manner and oriented towards functionality.

2. Mission control interface

The ARSI software system is based on the Robotic Operating System (ROS) architecture, which allows software modules to seamlessly communicate and exchange data on a local network. ROS includes various visualization tools, in particular a Qt-based visualization tool called RVIZ.

RVIZ is an open-source tool designed to be highly configurable, allowing users to develop their own panels and tools which are then displayed in the main interface. In phase II of this project we dedicated significant effort into developing a Mission Control interface based on RVIZ. The current Mission Control interface and its main panels are shown in Figure 1.

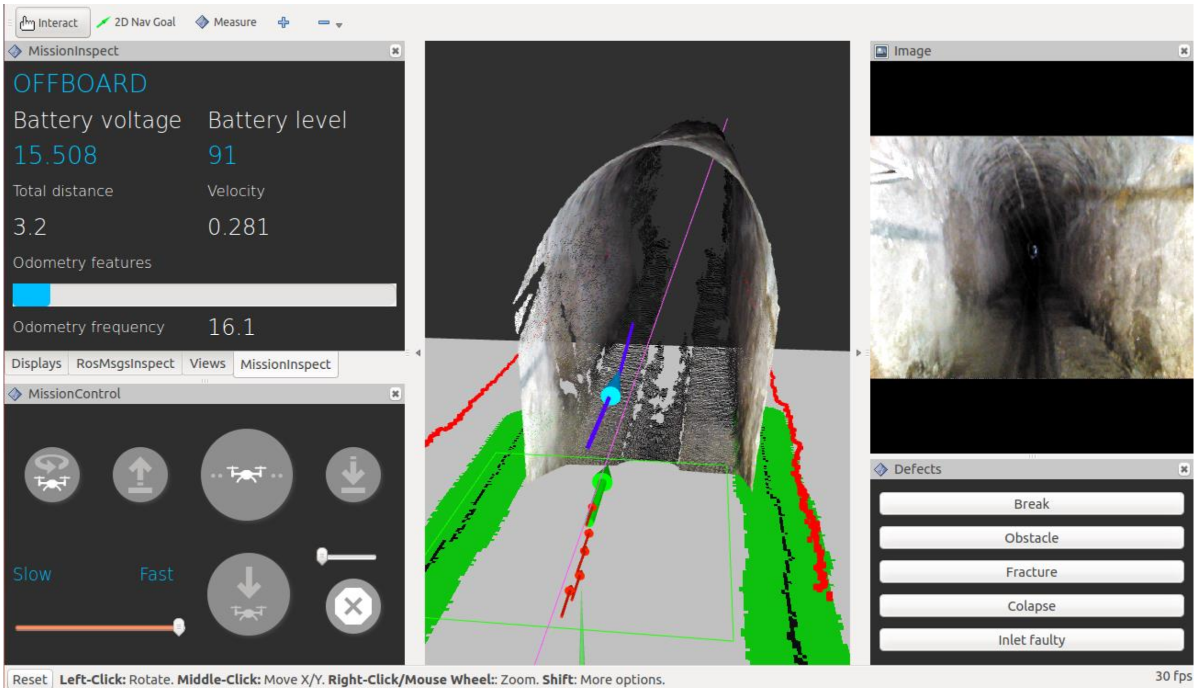


Figure 1: ARSI Mission Control interface

Mission monitoring

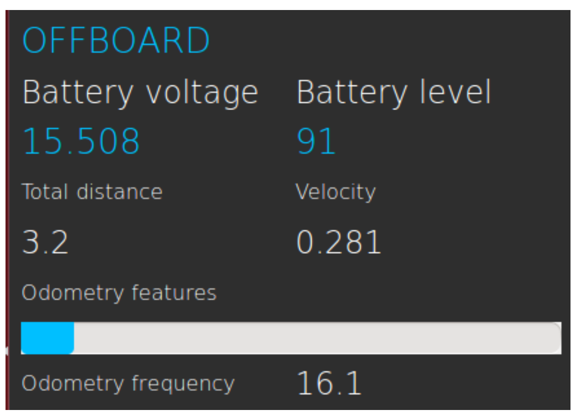


Figure 2: Mission monitoring panel

This panel provides a summary view of the critical information required to operate the ARSI MAV: battery levels, distance travelled, MAV velocity, and odometry quality.

Camera view

The camera view displays a live feed from the on-board camera. Due to bandwidth restrictions, the live image is down sampled, both in resolution and frequency. In our tests we found that at grayscale VGA image at 5Hz provided sufficient feedback for flight operations.

All sensor data is recorded on-board at full resolution for post-processing.

Mission control

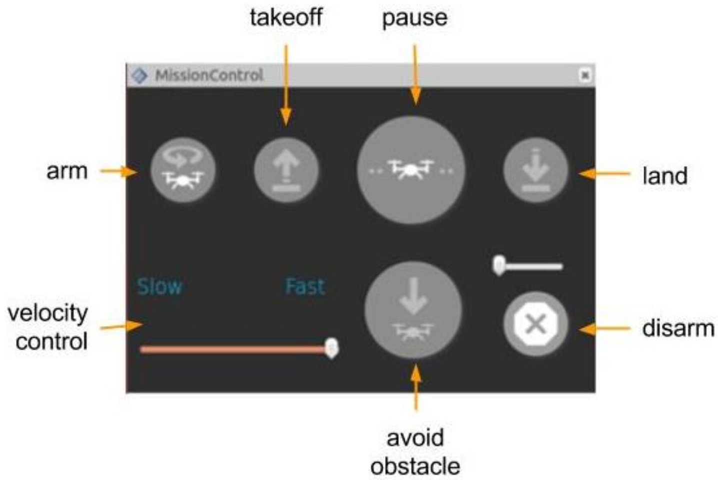


Figure 3: Mission Control panel

This panel contains all the MAV operation controls currently available to the operator. The controls allow operators to start the motors (arming), and start an inspection mission (including take-off). During the flight, operators can pause the MAV (e.g. to observe a point of interest), adjust its altitude and velocity, avoid obstacles, land, and finally disarm.

Note that since the only obstacles present in Mercado del Born are metal pipes at mid-altitude, avoiding obstacles currently consists in the MAV lowing its altitude to fly underneath them. In phase III, we would develop a dynamic obstacle avoidance scheme based on sensor information, in particular from the RGB-D camera.

We expect that new control modes will be developed in phase III, as we gain experience from operations with the ARSI MAV for sewer inspection. For example, the concept of operations presented in [2] showed that the ability to fly backwards could simplify mission planning in certain situations.

Defect entry

This panel is designed to allow operators to take note of a defect in the sewers during a flight. Since the MAV has a limited battery autonomy, pausing for several minutes to observe and record defects is not a viable solution. As described in our concept of operations, detailed defect detection is done using the analysis tool presented in [4].

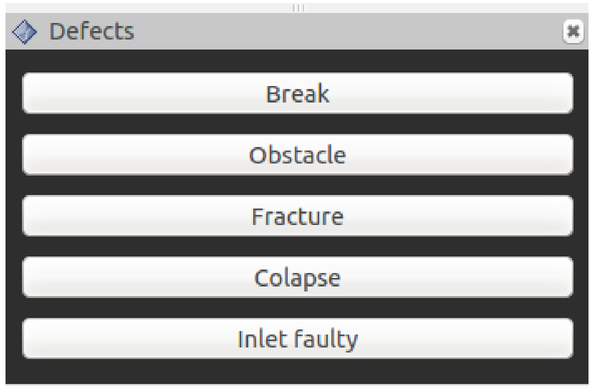


Figure 4: Defect tagging panel

However we wanted operators to be able to make note of any point of interest seen during the flight. This panel contains buttons for the 5 most common defects: breaks, obstacles, fractures, collapse and faulty inlets.

Flagged defects are timestamped and sent to the ARSI MAV, to be recorded in the data file generated on-board. All flagged defects are highlighted in the ARSI app timeline (see Figures **¡Error! No se encuentra el origen de la referencia.** and following). This way, experienced operators can flag most of the relevant information during the flight, pausing the MAV for a few seconds if necessary, and review this information in post-processing.

3D view

The 3D view is mainly an engineering tool but it provides essential information about the MAV during flight:

- Estimated MAV pose
- Visual odometry trajectory
- Mission global path
- Local planner trajectory
- Real-time position and yaw setpoints
- Laser data
- Costmap
- RGB-D point cloud
- Range measurements from the TeraRanger and ground estimation

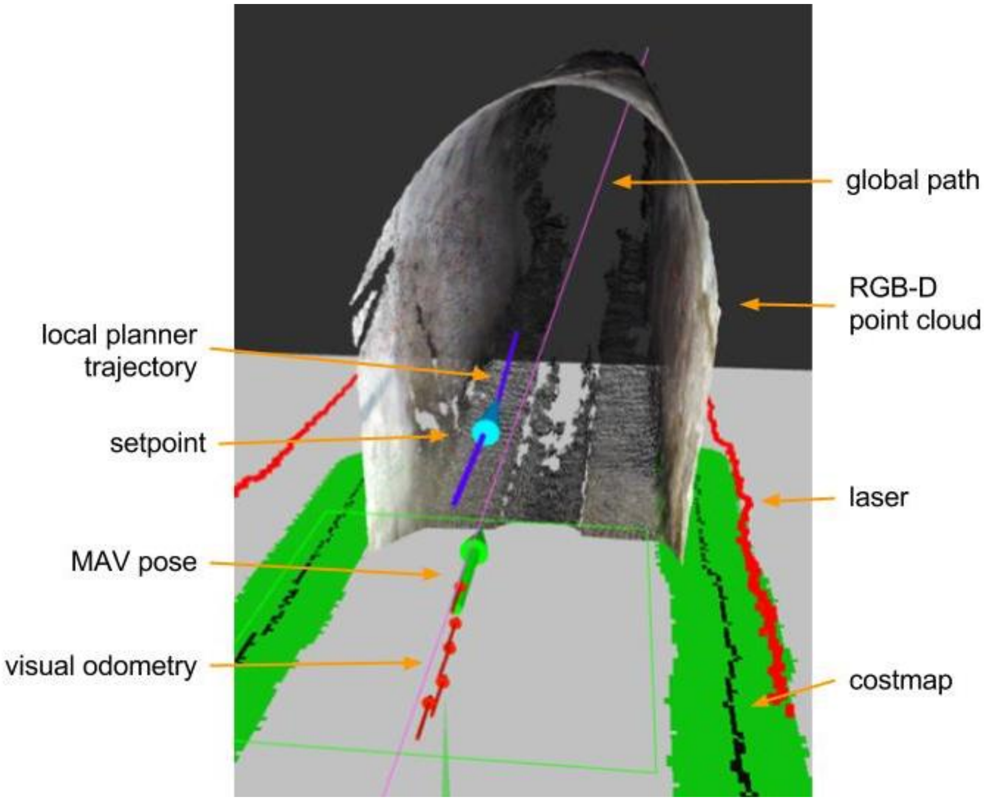


Figure 5: 3D view

3. Processed data in MAV missions

The main type of data processed and delivered during missions are images, video and depth maps. Since the amount, type and format of data has evolved along the development of the autonomous navigation of the MAV, here it is described briefly the evolution of the hardware chosen for recording imagery, since the beginning of phase II.

2.1. Evolution of the image sensor on the MAV

At the end of phase I the design of the MAV involved 4 fish-eye HD cameras placed at each side of the platform [1], so that the whole section was visually scanned. All this information was available for offline defect inspection and 3D mapping. A significant effort was made to integrate the image sensors on the MAV, in terms of hardware and software. Data in a work environment was acquired and the first developments towards 3D reconstruction were made. However, as the work and research on autonomous navigation progressed, it soon became apparent that the MAV needed more reliable sensors for estimating visual odometry and mapping its surroundings, as well as a more powerful on-board computer. As explained in [4] first a RGB-D camera (VGA resolution, 60º HFOV) was added, along with a better (and heavier) embedded computer, Intel NUC i7. The constraints on weight mandated to remove mandated that certain non-critical elements be removed as sensors for navigation were added, until weight-related issues are addressed in phase III. Currently the image sensor installed on the platform is the RGB-D. This layout (only one camera of VGA resolution and limited HFOV) impedes in phase II the capability of the MAV of performing defect inspection, but at the same time provides ARSI software with precise depth information that can be used to generate 3D models of the sewer traversed. It is our intention to keep optimising the components of the platform as the design of off-shell hardware improves so that the payload of the MAV may be increased or changed, and more image sensors may be added for visual inspection. As it is detailed in [4], it was necessary to make some compromises in the design of the MAV to attain autonomous navigation, our goal is to return to the sensors chosen in phase I design.

2.2. Data acquisition

As mentioned before, in a mission the ARSI system stores on-board a set of relevant data, amongst which the following list is used for post-processing:

- an image sequence recorded at 30 Hz,
- a sequence of depth maps generated by the RGB-D
- images flagged by the operator as defects (see section 2) and
- a sequence of poses produced by the odometry methods used by ARSI for its navigation.

This data is transferred to a computer dedicated to post-processing when the drone is taken up to surface, according to the operability described in [2]. Once the data is off-board it is processed to be used by the visualization interface (see **¡Error! No se encuentra el origen de la referencia.**).

2.3. Data processing

The tasks performed during the data processing stage are divided in two types:

- Tasks related to imagery.
- Tasks related to the generation of a 3D model

The resulting files from this step are saved in a directory tree in this manner:

```
Arsi_2017-09-14-12-20-17
├──report_data
├──modelo_3d
│   └──mesh
└──images
```

Figure 6.- Output directory tree from data processing

As Figure 6 shows, each mission is stored in a folder with the name corresponding to the data when the mission was executed. The snapshots of defects observed are stored in the folder "report_data", with attached data to account for when the defect was seen. Both the 3D model and the sequence of images are stored in their corresponding folders.

Imagery

In this step a 30 Hz video (VGA resolution) is produced from the image sequence, along with a list of timestamps for each frame. This list of timestamps will be used by the visualization app for synchronizing the video and the flagged images with the display of the 3D model. In other words, the timestamps of the frames are the link between the instant when a defect was seen (image sequence) and the location of the defect (3D model), since each camera pose in the 3D model is associated to a timestamp.

Figure 7.- Stored image taken from the MAV on a mission

A video generated from a sequence of images taken by the MAV on a mission can be watched in [7].

3D model

In order to build a 3D model of a scene it is compulsory to have a precise and tuned estimation of the camera poses. In order to obtain real time pose estimation information, ARSI makes use of visual odometry modules of RTABMAP, a RGB-D Graph-Based SLAM approach (see [4]). However, the camera poses estimated by the visual odometry modules of RTABMAP are not accurate enough as to be used in the offline generation of dense 3D model of the sewer transited.

We overcame this problem by applying methods from OpenMVG[1], an open-source library specialized on structure from motion problems. OpenMVG manages to reduce significantly the pose estimation error on sequences of over 1000 images in a few minutes. This is achieved

[1] https://openmvg.readthedocs.io/en/latest/

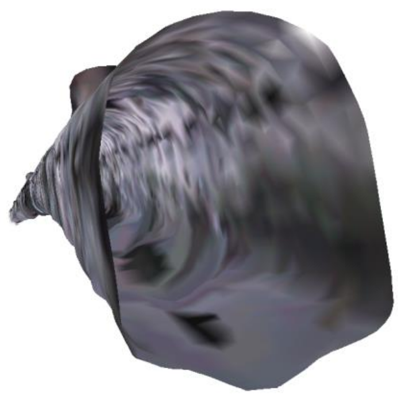
despite the amount of dust and particles that are lifted by the propellers of the MAV[2] (see Figure 8).

Figure 8.- Typical image with dust.

In order to make the most the data produced by MAV in the mission, the next natural step after refining the camera poses estimation is to integrate the depth maps with the new camera poses and generate a 3D model of the mission[3]. This step will be implemented in the early stage of phase III. In phase II we have implemented a dense 3D reconstruction by means of MVE[4], an open-source 3D dense reconstruction software. Figure 9 shows the differences between a 3D model computed with camera poses not optimised and a 3D model camera poses previously optimised and refined.

[2] The link provided in [8] takes to a video sequence that shows the amount of dust present in the air.
[3] Additionally, the use of depth maps will help to resolve the problem of dust, since the generation of these maps is more robust to dust than optical images.
[4] https://www.gcc.tu-darmstadt.de/home/proj/mve/

Without camera optimisation



With camera optimisation

Figure 9 Visual comparison between 3D model without optimised pose cameras (left) and 3D model with optimised pose cameras (right).

Eurecat, as technological partner of the consortium, has a in-house 3D dense reconstruction software system that offers excellent results and has been validated in a variety of environments. We have tested this software with imagery from sewer missions with promising results. However this option has been left aside for the moment since the RGB-D camera already produces one depth map per frame and it would be inefficient not to use this information. We are exploring the possibility in the future of combining the depth maps available and the 3D dense and meshing methods from Eurecat to produce high-quality, efficient 3D models of sewers.

The video given by [9] shows briefly the 3D dense reconstruction attained by optimizing first the camera poses estimation and subsequent dense reconstruction using open sources. The sequence is made up of 1000 frames and covers a stretch of 60 meters. The mission lasted 5 minutes.

The goal of the data processing taken on the missions is to allow the sewer operator to assess the serviceability of the sewer and assist the defect inspection. Next section discusses how we have achieved this.

2.4. Discussion on processed data.

The visual information produced, and especially the 3D model created out of material collected on a mission should allow the sewer operator to analyse geometrically the contours and volume of the sewer, and perform structural defect inspection. In particular, we understand that the 3D model should allow to:

- Assist the inspection of structural faults, such as: cracks, fractures, breaks, breaks with loss and collapses
- Supervise the serviceability of different types of sewer.
- Record HD, 30 Hz videos and take snapshots at sewer operator command.
- Sewer elements recognition of: manholes, inlets, intersections, section change, changes of slope.
- Comparison of obtained data with reported information of the sewer (type and section)
- Sewer map building.
- Distances measurements.

In this phase our consortium has already informed that the task of automatic assistance for sewer inspection will be addressed in phase III, and that in phase II we would focus on developing autonomous navigation and 3D mapping. All in all, we believe that the quality of 3D reconstruction achieved provides a functional system that enables sewer operators to visually inspect the sewer with the offline visualization app and tag structural defects and faults found in the 3D model. The operator is able with the app to tag a notorious failure seen in the video or/and the 3D model that might have been overlooked during the mission, take a snapshot, tag it, extract the distance from the beginning of the 3D model and make a report with the whole set of incidences seen during the mission and offline.

The consortium has acquired an HD RGB-D camera, and both its hardware and software have been adjusted to work on the MAV. The HD camera will be validated and functioning during the first stages of phase III. Needless to say that HD images will significantly increase the quality of the imagery and the 3D models produced.

4. User Interface for data visualization

A Usability and Graphic design process has been developed in order to assimilate the look and feel of the data generated by ARSI software.

4.1 GUI overview

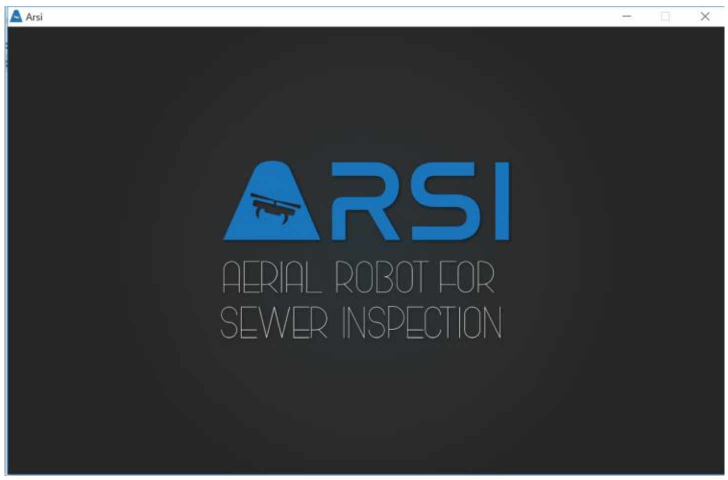


Figure 10.- ARSI splash screen

The new Offline App developed in Phase 2 is the tool to review the inspection data easily and edit the defects detected during the navigation stage of each mission.

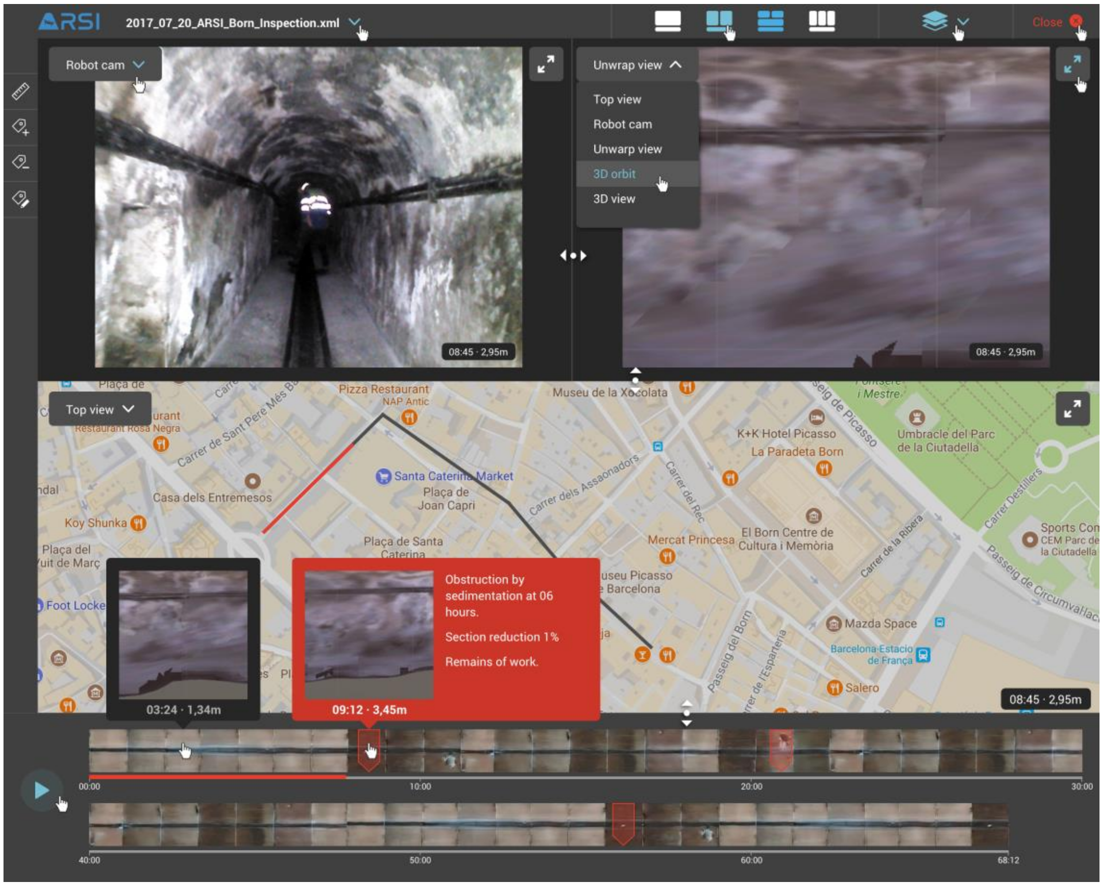


Figure 11.- Initial Arsi Offline Viewer and Editor App GUI / Ux Proposal

Even though it is our understanding that the best way to assimilate the visualisation app is to see it working and to actually use it, here we briefly describe its functionality, settings and behaviour. Different versions of each screen layout have been developed and discussed, and can be reviewed at [10]. For a comprehensive overview of how the visualization app works the reader is referred to the video at

The app presents a layout with four main areas: the video, the 3D model, the map where the mission area is done, and a timeline that notifies of the images where a defect was flagged. All the areas are linked so that if the cursor moves along the timeline, the video will move along to keep up with the timeline. Likewise, the 3D model will move accordingly, as well as the cursor along the path of the mission on the map.

The virtual camera used to navigate the 3D model can be oriented in any direction (similarly to Google StreetView) to inspect the ground, vault or walls of the sewers. This includes a virtual drone to show where the camera pose of the drone is located at any time.

4.2 Offline Viewer

The app counts with layout templates and configurable viewports so the user can set different screen setups:

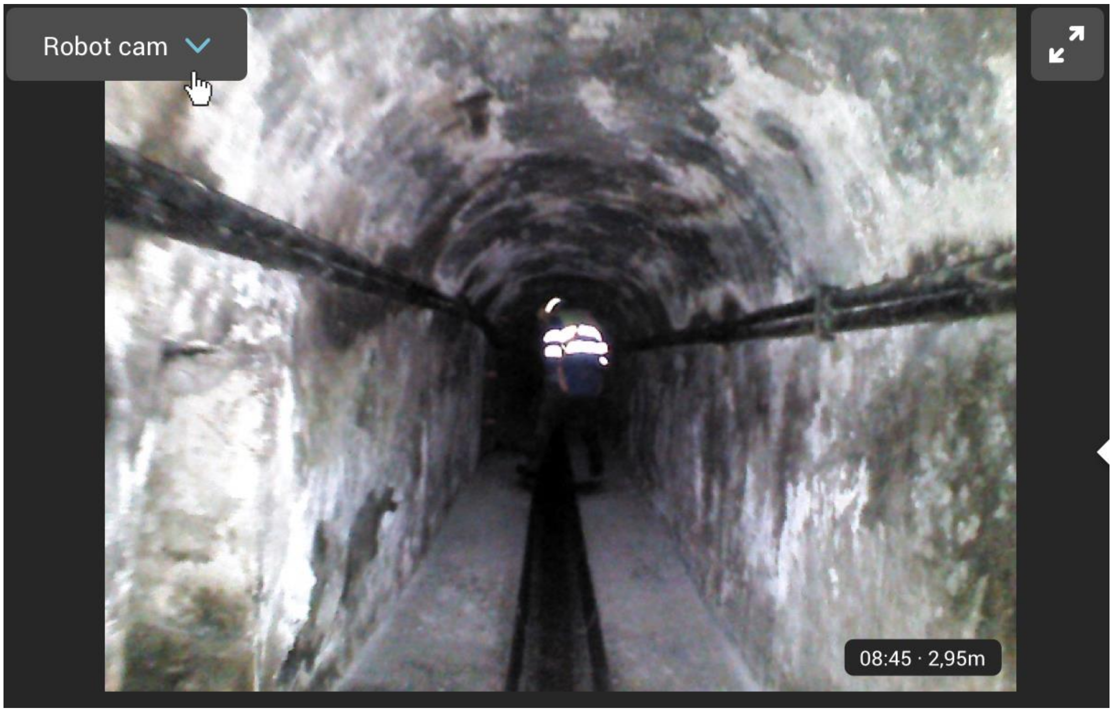


Figure 12.- Drone camera View

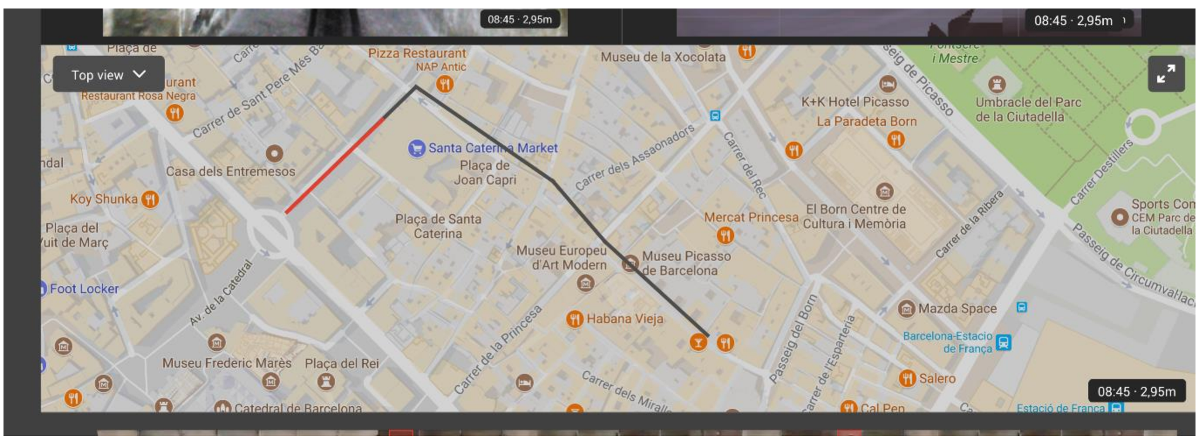


Figure 13.- Top and Carthography view

Figure 14.- 3D reconstruction 360º camera (work in progress)

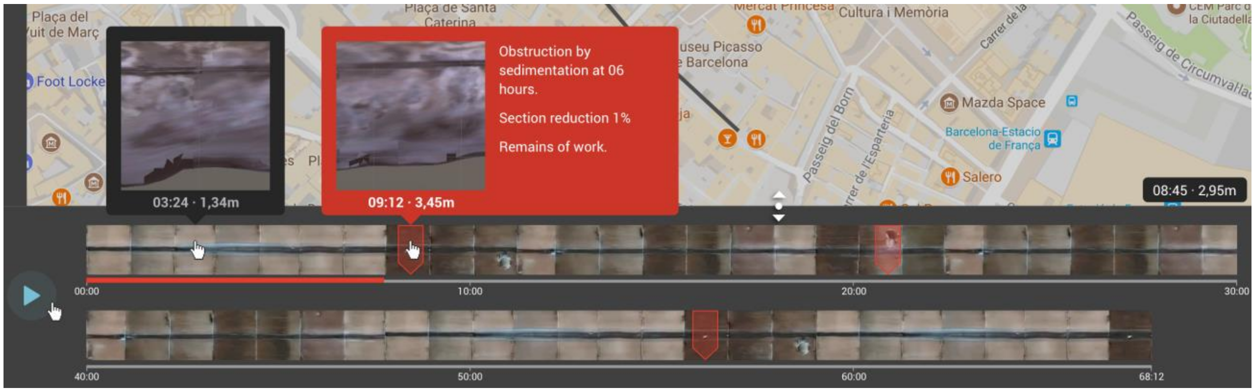


Figure 15.- Scrubbing time line. Access to Report Data on Defects

The Timeline shows each photogram of the mission by hovering the pointer of the mouse along the sequence.

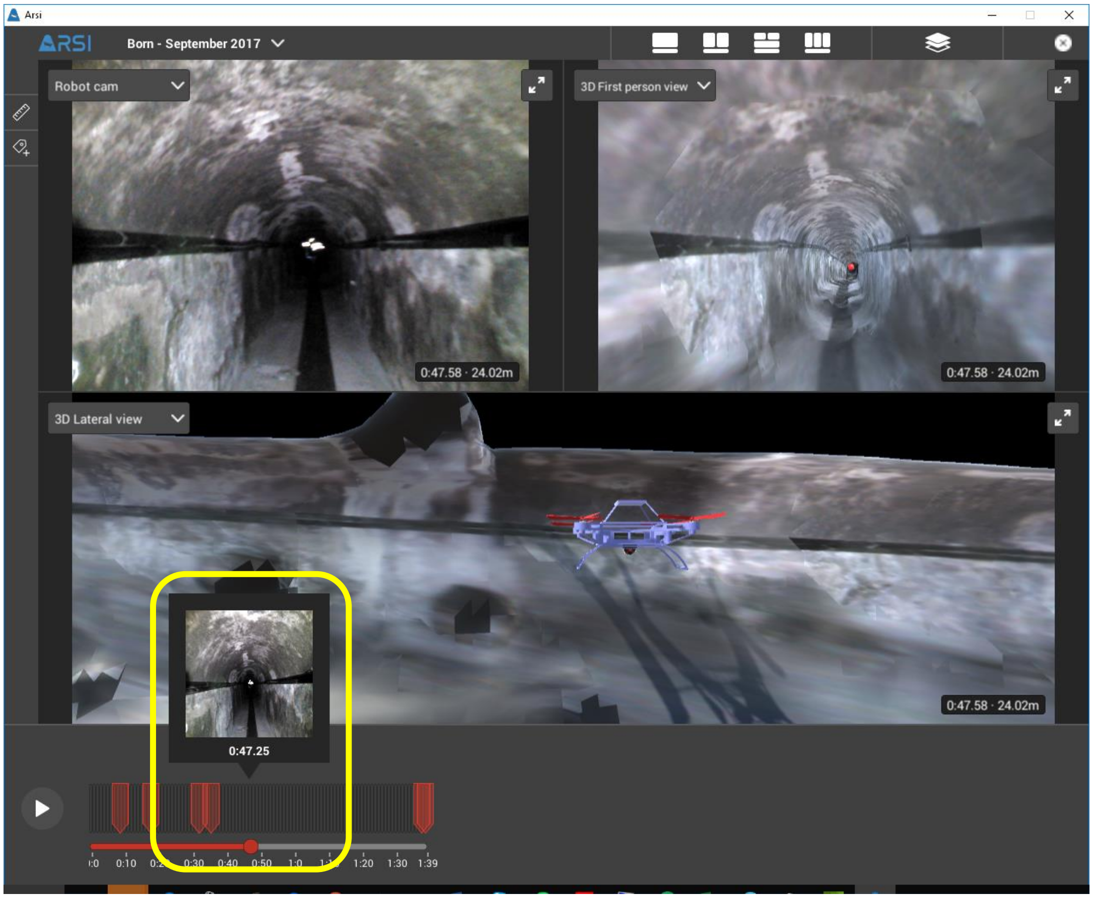


Figure 16.- Hover on timeline to view video frame.

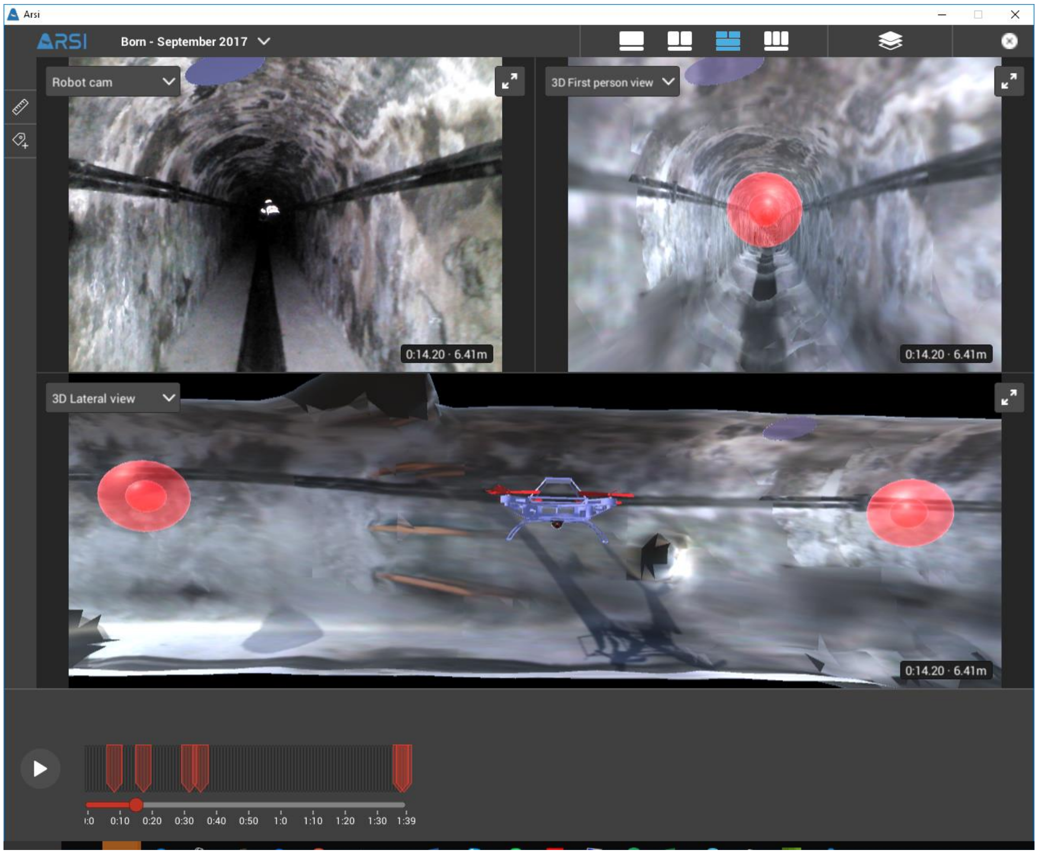


Figure 17.- Mission Time Line with Drone cam / 3D reconstruction / side cam

4.3 Defects/ Editor features

The same functionality allows to have a thumbnail of the defects tagged during the navigation stage, allowing the user to edit defects information, delete false detections, or adding manually other defects detected in the offline tool.

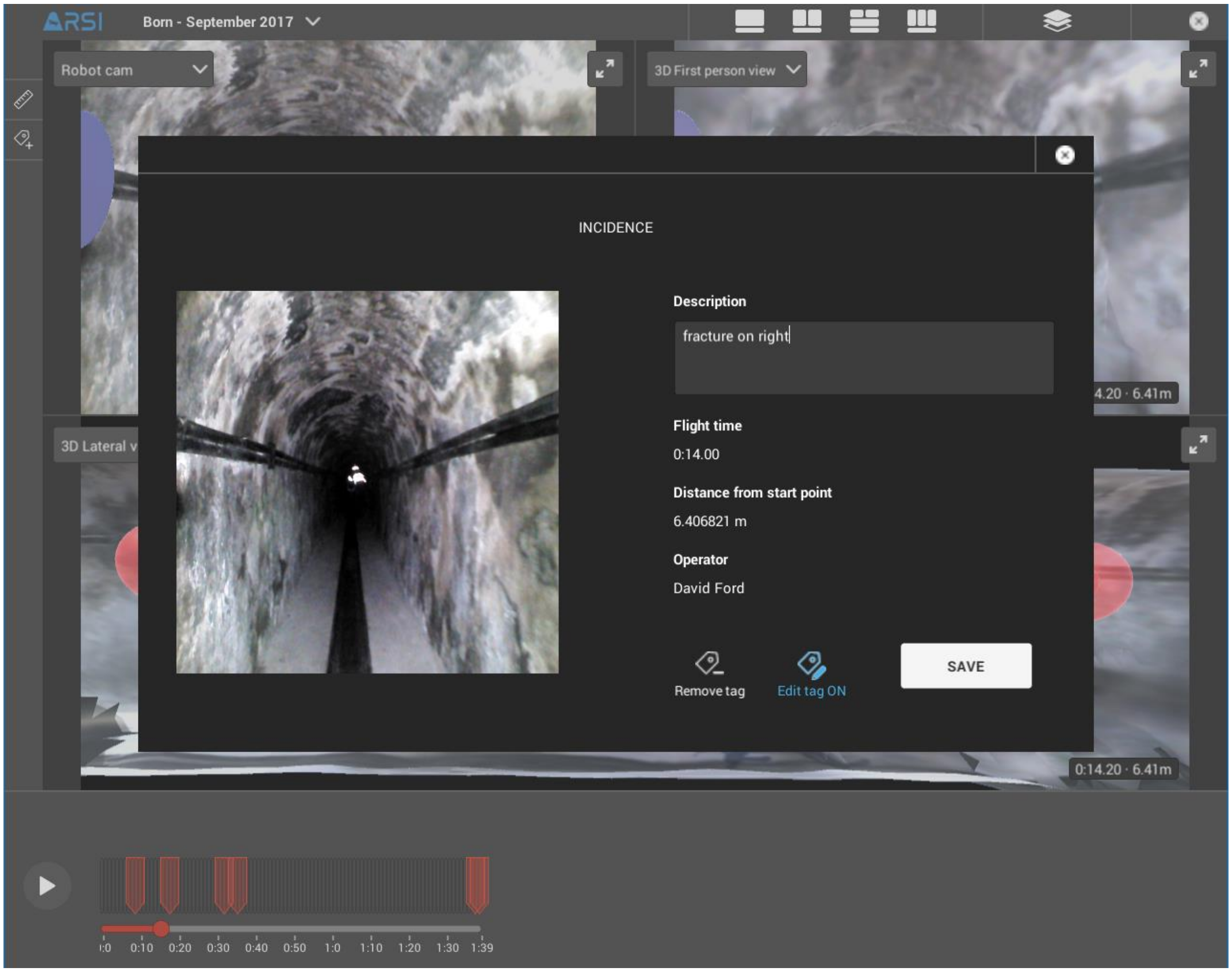


Figure 18.- Defects Tag Editor

Defects detected offline allow to edit the description and save the time stamp and distance from origin along with a snapshot of the Drone Cam.