METRICS 2nd RAMI CASCADE CAMPAIGN

Aerial Domain

Competition description

FADA-CATEC
Francisco J. Pérez Grau
Pablo León Barriga

Contents

Abstract

A complete description of METRICS' RAMI 2nd Cascade Campaign regarding the aerial domain, in conjunction with ICUAS 2023. Aspects as input data given to teams, results and data expected from participants and evaluation metrics are addressed.

1 Competition overview

The Cascade Campaign in the aerial domain will consist of two different Functional Benchmarks (FBM), corresponding to two of the benchmarks for ICUAS'23 UAV Competition.

The first one, also referred to as FBM1, is related to robot localization without GNSS. This challenge addresses the problem of precisely estimating the drone position and orientation using only data from on-board sensors.

The second one, or FBM2, consists of the automatic detection of defects using advanced AI algorithms. This challenge addresses the problem of finding AI methods to detect where a defect is in a given image, and data handling in order to augment and diversify a small dataset to achieve a good performance in completely new scenarios.

As the Cascade Campaign is meant to be carried out online, a Docker image will be provided to teams. This image will contain the necessary data for each FBM, a dummy example of participation for both FBMs, as well as useful *readme* files explaining the distribution of the files in the image and the expected results.

The competition has an evaluation period, during which teams can upload their provisional results in order to estimate their current standing in the competition. More details about this online scoreboard are described in section **??**.

2 Input data

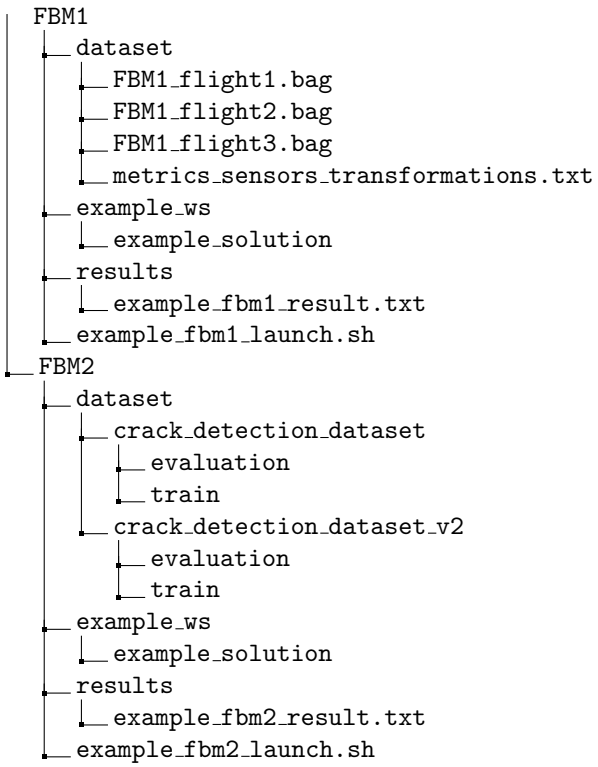
Across this section, the data available for every participant will be detailed.

2.1 Docker image and practical guide

To ensure an equal starting point for everyone, a Docker image will be released containing some relevant dependencies installed, such as ROS. Also, some repositories will be already installed, such as CUDA and CUDNN, but you must reinstall them on your machine to avoid runtime errors.

The Docker image is based on Ubuntu 20.04 LTS. Over the 'metrics' personal folder, we can find the following structure:

Figure 1: Docker image personal metrics folder overview.



For each FBM, there is a folder containing its dataset, an example workspace (example_ws), a results folder, and finally an example of a bash script.

Hereinafter, the datasets available will be expanded.

2.2 Datasets

2.2.1 FBM1

Regarding the first FBM, the data provided consists of three ROS bag files. Those bags are allocated in ***/home/metrics/FBM1/dataset/***, and they belong to three different flights, where each one corresponds with a gradually increasing level of difficulty. Starting from an easy trajectory without relevant heading changes, passing through another easy trajectory with heading variations along the flight, and ending with a more complex trajectory with heading variations as well.

Every bag file contains the same type of messages at the same rate, although they have different durations and thus, different number of messages. The information contained in those bags come from four different sources:

- **RealSense D435i camera sensor**: providing both color (RGB) and depth images at 30 Hz, as well as the camera intrinsic parameters through *CameraInfo* topic, accelerometer measurements at 250 Hz and gyroscope measurements at 200 Hz.
- **DJI A3 Autopilot**: providing IMU data at 400 Hz.
- **VICON Motion Capture System (Ground-truth)**: a pose with the actual position of the drone with millimetre-precision at 100 Hz. It should be noted that this source is only available in the provided bags, but it won't be available in the final evaluation dataset, **so don't use it for your algorithms, only for auto-evaluation**.

The transformation between the ground truth pose and the sensors is described in *metrics_sensors_transformations.txt* file, which is inside the FBM1 dataset directory. As mentioned above, the information will be dealt as ROS messages in a ROS bag file. The message types used are:

- **/camera/color/camera_info**: *sensor_msgs/CameraInfo*
- **/camera/color/image_raw/compressed**: *sensor_msgs/CompressedImage*
- **/camera/depth/camera_info**: *sensor_msgs/CameraInfo*
- **/camera/depth/image_rect_raw**: *sensor_msgs/Image*
- **/camera/accel/sample**: *sensor_msgs/Imu*
- **/camera/gyro/sample**: *sensor_msgs/Imu*
- **/dji_sdk/imu**: *sensor_msgs/Imu*
- **/vicon_client/METRICS/pose**: *geometry_msgs/PoseStamped*

2.2.2 FBM2

Regarding the second FBM, a cracks defect dataset is provided. This dataset contains up to 11.298 images labelled with YOLO annotation format[1]. From the total images, 9.844 images contain at least one crack, and 1.454 images don't. The full dataset is divided into two folders: one with 9.603 images for training, and the remaining 1.695 are for validation.

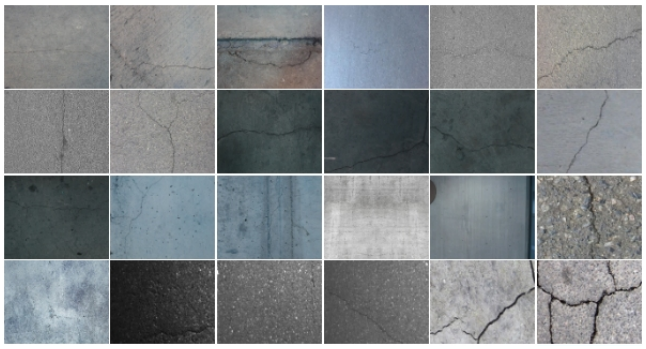
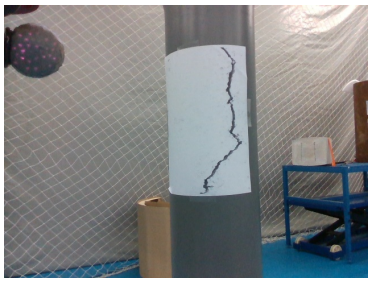


Figure 2: Dataset images sample.

The images' size is 448x448 px compressed in JPG format. It is **important** to note that the images that will be used for evaluation of the FBM2 will be somewhat different from the images of the dataset, meaning that the images will be completely new images of cracks on some surface or images extracted from the dataset, printed and placed over different locations as shown in Figure 3(b).

(a) Image not contained in the dataset.



(b) Printed image from the dataset.

Figure 3: Example of possible images for competition evaluation.

Thus, it is highly recommended to perform data augmentation over the crack defect dataset in order to succeed with the final evaluation.

[1] `https://github.com/AlexeyAB/Yolo_mark/issues/60`

3 Required data

Now it is time to talk about how the teams must provide their data in order to be evaluated. **Those teams that do not strictly meet the requirements described over this section (3.1) won't be evaluated**.

3.1 Final evaluation

First of all, every team must submit their Docker image as a Docker hub repository. Please refer to *Metrics Docker Guidelines* subsection 3.3 if you have never done this before. The submitted Docker image, in turn, has to fulfill the following requirements:

- **Contain the end to end solution**. This means that it needs to contain every single file needed to run the solution from the dataset format (ROS bag file or images folder) to resulting txt file (mentioned below).
  For each FBM, this end to end solution must be placed over a *team_name_ws* folder. If my team were CATEC, our workspace will be named *catec_ws* for example.

- **Contain every dependency installed and documented**. Every single dependency that was not installed in the provided Docker image must be installed and documented (download link/apt package name, and installation guide if it is non trivial) in a *dependencies.txt* file inside the *team_name_ws* folder.

- **Running scripts**. For each FBM one bash script, as the example given, must be placed over the corresponding *FBMx* folder. The scripts must take as arguments: the evaluation data and the resulting txt filename (as the example one). Then, running the script should create the results .txt file over the *FBMx/results* folder. This .txt file will be used to evaluate the team's performance according to section 4.

- **The result must fit into the evaluation requirements**. The most important aspects will be discussed in section 4, but in case of specific doubts, please refer to *Evaluation Plan*'s document[2].

[2] `https://metricsproject.eu/wp-content/uploads/2021/05/METRICS-RAMI-Evaluation-Plan.pdf`

4 Evaluation summary and results format

Across this section, the evaluation metrics used for both the preliminary and final evaluations will be commented, together with the format of the .txt files to be evaluated.

Further information about the complete evaluation procedure is widely described in the *Evaluation Plan* document (3.1), subsection 4.2.

4.1 FBM1

For the FBM1, the required output file is referred in the *Evaluation Plan* as the ***Pose Data***. This data must be formatted as a plain text file with one pose on each new line, with the following information separated by blank spaces:

- **Timestamp** (since Epoch)
- **Translation X** coordinate (m)
- **Translation Y** coordinate (m)
- **Translation Z** coordinate (m)
- **Orientation X** component (quaternion)
- **Orientation Y** component (quaternion)
- **Orientation Z** component (quaternion)
- **Orientation W** component (quaternion)

The file must be named according to the second input parameter of the running script.

For example, one txt line with pose data could be:

```
#timestamp tx ty tz qx qy qz qw [This line should not appear in your result]
1403636580.013555527 0.0125827899 -0.0015615102 -0.0401530091 -0.0513115190
0.8092916900 0.0008562779 0.5851609600
```

Note that the minimum acceptable rate for evaluation is 1 Hz, but a higher rate is recommended as the evaluation will be performed against the VICON pose at 100 Hz, so low rates will lead to higher errors.

The metric used to obtain the FBM1 score will be the RMSE (Root-Mean-Square Error) in rotation and translation, and this error will be obtained using the EVO package for evaluation of odometry algorithms[3].

[3] `https://github.com/MichaelGrupp/evo`

$$RMSErrors = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}$$

Figure 4: Mean-Root-Squared Error formula

4.2 FBM2

For FBM2, the required output file is referred in the *Evaluation Plan* as the ***Detection Data***. This data must be formatted as a text plain file with one detection on each new line, with the following information separated by blank spaces:

- **Image name**
- **Top X** coordinate (px)
- **Left Y** coordinate (px)
- **Bottom X** coordinate (px)
- **Right Y** coordinate (px)

The file must be named according to the second input parameter of the running script.

For example, one txt line with detection data could be:

```
# image_name left top right bottom  [This line should not appear in your result]
frame0005.jpg 213 144 315 172
```

If one image contains more than one detection, each detection will be written in a new line with the same image name.

The metric used to obtain the FBM2 score will be the CSI (Critical Success Index) (see Figure 5).

$$CSI = \frac{TP}{TP + FN + FP}$$

Figure 5: Critical Success Index formula.

To determine which detection is a True Positive (TP), False Negative (FN) or False Positive (FP), the IoU (Intersection over Union) will be used. Figures 6 and 7 show their definitions.

The detection satisfying that the IoU calculated with the original labels is

greater than a given threshold (IoU >= 0.4) will be counted as a True Positive. The detection that doesn't satisfy this condition with any of the image labels is counted as a False Positive, and the labels with no detections assigned are the False Negatives.

Figure 6: Intersection over Union formula.

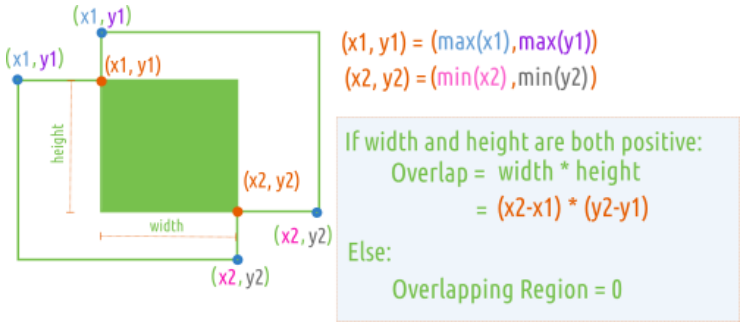


Figure 7: Overlapping region formula.

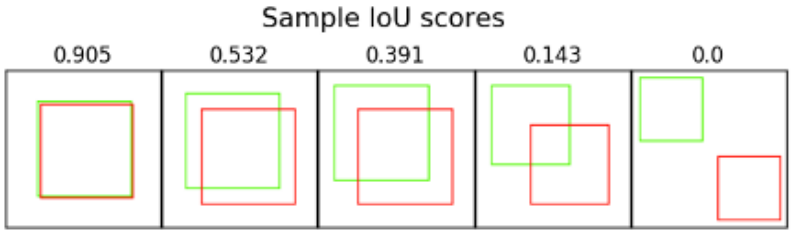


Figure 8: Intersection over Union examples.

5 Contact information

Official information concerning rules, interpretations, and information about the competition can be found on METRICS website[4], or you can contact us at im@metricsproject.eu.

[4] https://metricsproject.eu/inspection-maintenance/