1 Introduction

This document provides instructions for using the referee box and rosbag recorder for coordinating and recording data during benchmark executions in the HEART-MET competitions in Metrics. Also included are the steps for executing the benchmarks for the organizers.

2 Refbox

This section describes the packages that comprise the refbox and refbox client which are used for communication are recording benchmark trials.

Four ROS packages are provided, namely:

- `rosbag_recorder`: This package consists of a node with an interface to start and stop recording ROS bagfiles. The topics to be recorded are specified in a `.yaml` file. This node is expected to run on the robot, and the bagfiles will be stored on the robot.

- `metrics_refbox_msgs`: This packages contains definitions of ROS messages which will be used to communicate commands, results of benchmarks etc. This package is required on both the robot and the computer running the refbox.

- `metrics_refbox`: This package consists of a node which provides a UI for setting up benchmark trial configurations, sending commands to the robot to start or stop a benchmark trial and to log the results of benchmark trials. This node must be run on a computer which is on the same network as the robot, and will use the ros master of the robot for communication with the robot.

- `metrics_refbox_client`: This package contains a node which is the 'client' counterpart to the `metrics_refbox`, and runs on the robot. It is responsible for relaying information between the robot and the refbox, and for starting and stopping recording of ROS bagfiles via the `rosbag_recorder`. In addition, this package also contains a node which returns mockup results for each of the benchmarks. This is meant for testing; teams can also look here to see how the result messages should be filled.

2.1 Rosbag recorder

1. Clone the `rosbag_recorder` package in the robot's catkin workspace and build it.

2. Specify the topics to be recorded in `ros/config/topics.yaml`

3. In `ros/launch/rosbag_recorder.launch` specify the `file_path`. This is the path on the robot's computer where the bagfiles will be stored. Make sure that the path already exists.

4. In `ros/launch/rosbag_recorder.launch` specify the `file_prefix`. The names of all bagfiles will be prefixed with this string. This can be the team name or another string that identifies the robot.

If you are recording a large number of topics, the `timeout` parameter in the launch file may need to be increased.

2.2 metrics_refbox_msgs

1. Clone the `metrics_refbox_msgs` package both on the robot and the refbox and build it.

3 metrics_refbox_client

1. Clone the `metrics_refbox_client` on the robot and build it

3.1 metrics_refbox

3.1.1 Setup

1. Clone the `metrics_refbox` on the refbox computer and build it

2. Set the `ROS_MASTER_URI` environment variable on the computer to point to the ROS master of the robot being tested (for example: `export ROS_MASTER_URI=http://192.168.1.100:11311` where `192.168.1.100` is the IP address of the robot).

3. If necessary, set the `ROS_IP` environment variable on the computer to its own IP address (for example: `export ROS_IP=192.168.1.200`, where `192.168.1.200` is the IP address of the refbox computer). It is only necessary to set the `ROS_IP` if the router you are connected to does not provide name resolution. You can test if this is required by using the 'Test communication' button in the GUI.

4. In the previous step, if you have set `ROS_IP` on the refbox, you must also set the `ROS_IP` on the robot. It should be set to the IP address of the robot.

5. Make sure that the `ROS_HOSTNAME` is set neither on the robot nor the refbox.

3.1.2 Benchmark configuration

The configurations for each benchmark are defined in `config/`. The `refbox.json` file contains the list of benchmarks and properties for each benchmark, such as the topic and type of the result message.

The configuration for individual benchmarks is in their associated `.json` file. This includes the possible variations for the benchmark, file where trial configurations are stored, and the name of the module and class where benchmark specific actions are performed. All benchmark-specific elements in the refbox GUI are generated based on these configuration files.

In `config/trials`, the configurations for trials for each of the benchmarks is stored. A trial configuration consists of an instantiation of the different variations in a given benchmark. These files are generated by the GUI, and their contents can be edited through the GUI as well.

3.1.3 GUI

The GUI consists of the following elements (see Figure 1)

1. Team selection: select the team currently running the benchmark

2. Communication test: test whether the refbox and refbox client on the robot can communicate with each other. If a reply is not received from the robot (seen in the Status box), there is a problem with the setup on either the refbox, the robot or both.

3. Benchmark selection: select the benchmark which is being currently executed. The configuration in 5., and the results in 7. will change according to this selection.

4. Trial configuration: select and edit configurations for trials for the current benchmark. A trial configuration consists of an instantiation of the configurations in 5. You can `Generate` a new random configuration and optionally edit it by first `Unlock`ing the trial configuration. If you edit it, you must click `Save` to save it. Trial configurations can be `Delete`d as well.

5. Current trial configuration: shows the configuration of the current trial. If this is edited, you must Save it to make the changes permanent.

6. Benchmark controls: The `Start` and `Stop` buttons send the corresponding command to the robot (via the refbox client). `Previous` and `Next` can be used to go to the previous or next trial.

7. Results: Once the robot sends back a result for a trial, some fields of the result are shown here. You can add additional notes to the result if necessary (in this case, you must click the `Save` button to save the notes)

8. Status box: various info and error messages are shown here

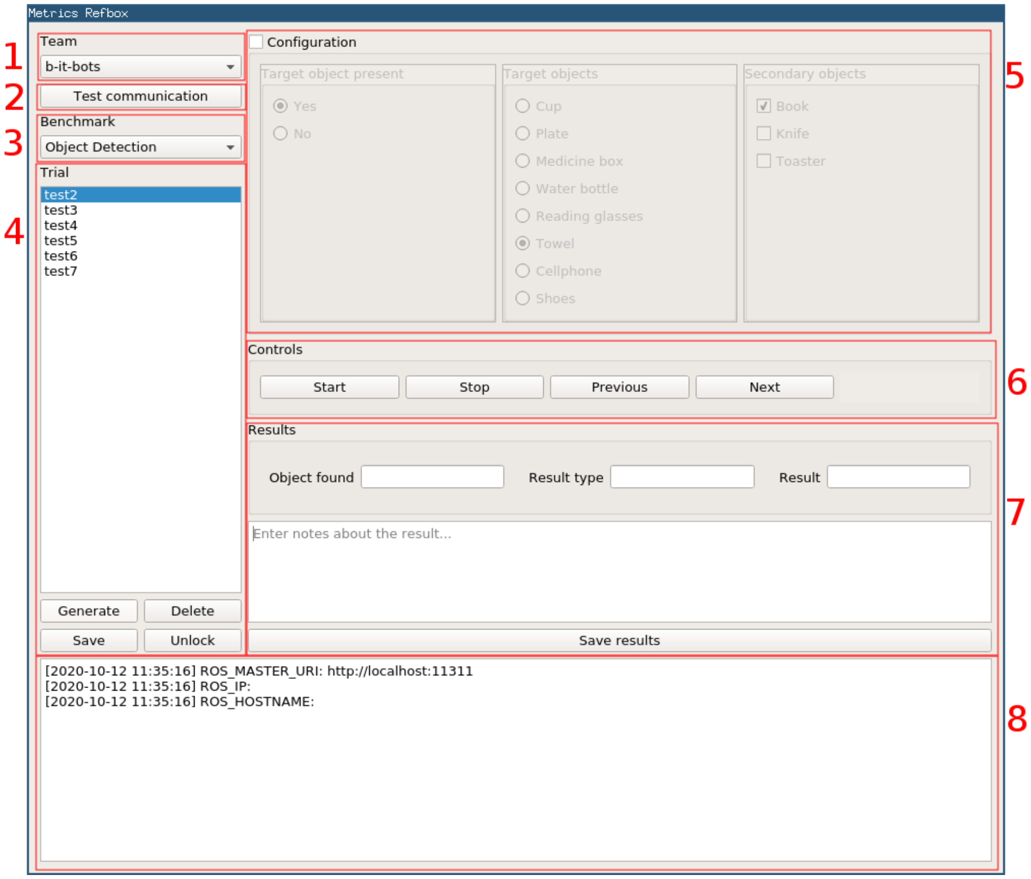
3.2 Usage

After completing the setup on both the robot and refbox, follow these steps to start and record benchmark trials.

1. On the refbox, launch:
   ```
   roslaunch metrics_refbox metrics_refbox.launch
   ```

2. On the robot, launch:
   ```
   roslaunch metrics_refbox_client metrics_refbox_client.launch
   ```
   (Note: this also launches the `rosbag_recorder`)

3. Test communication with the `Test communication` button on the refbox. A status message should be printed that you received a reply from the refbox client.



Team: b-it-bots

Benchmark: Object Detection

Target object present: ● Yes

Target objects: ● Towel

Secondary objects: ☑ Book

```
[2020-10-12 11:35:16] ROS_MASTER_URI: http://localhost:11311
[2020-10-12 11:35:16] ROS_IP:
[2020-10-12 11:35:16] ROS_HOSTNAME:
```

Figure 1: GUI

4. Select the team and benchmark to be executed.

5. Generate (and optionally edit) a set of trials to be executed. These trials should remain the same for all teams.

6. Select a trial and click `Start`.

7. Once the start message is confirmed by the client, a timer will begin.

8. In case the start message is not confirmed within a timeout (around 10 seconds), recheck whether communication with the robot and whether the rosbag recorder is running on the robot.

9. When the robot completes the trial, the results should be displayed. Add notes to the result if necessary (for example, if there were penalizing behaviours by the robot), and save.

10. Click `Next` to proceed to the next trial and continue again with step 6.

11. When all the trials have been executed, collect the bagfiles from the robot, and collate them with the results stored in `results/`

4 Competition instructions

This section contains instructions for organizers before and after execution of the benchmark trials.

4.1 General instructions

4.1.1 Team

1. Set up topics to be recorded. This is based on the benchmark being executed and the available sensors on the robot. See instructions for each benchmark below for more details.

2. Start refbox client

4.1.2 Organizer

The conditions in which trials are executed must be varied as far as possible. For example, trials should be conducted under different lighting conditions, time of day, locations around the test-bed, with different persons and with persons on different days, viewpoints and distances to the target of interest etc. Some of these variations are already captured explicitly in the trial configuration.

- For the given benchmark select the instances for particular variations in that benchmark. For example, select the target and secondary objects for Object Detection, or the target persons for Human Recognition
- Update the refbox config with these instances (see Section 3.1.2)
- Generate a set of trials for the benchmark via the refbox (see Section 3.1.3)
- For tasks which involve robot or human motions, set up an external camera (or multiple) which captures both the robot and the scene/human that it is interacting with
- Make sure the refbox computer is on the same ROS master as the current robot (see Section 3.1.1)
- Start refbox and test communication with the robot
- Start recording with the external camera
- Select the team, benchmark and the first trial and send the start command (more details in Section 3.2)

4.2 Object Recognition

The list of objects to be used for this benchmark are listed below. If a source is listed, it is only a suggestion. Any variation of the object can be used for the benchmark (multiple variations are encouraged). The sources listed as numbers are IKEA product IDs.

Object	Source	Datasets
Cup / Mug	102.773.66	ERL[1], YCB[2], Washington RGB-D[3]
Plate	302.589.13	ERL, YCB, Washington RGB-D
Bowl	804.239.77	YCB, Washington RGB-D
Towel	003.536.19	Washington RGB-D
Shoes		
Sponge	602.576.05	YCB
Water bottle		ERL, Washington RGB-D
Toothbrush		Washington RGB-D
Toothpaste		Washington RGB-D
Tray	104.199.50	
Sweater		
Cellphone		ERL, Washington RGB-D
Banana		YCB, Washington RGB-D
Medicine bottle	DE[4], UK[5], IT[6]	
Reading glasses		ERL
Flashlight		Washington RGB-D
Pill box	DE[7], UK[8], IT[9]	

Table 1: List of objects for Object Detection

The topics to be recorded are:

1. *RGB image(s) from camera(s)

2. *Intrinsic camera calibration

[1] https://www.eu-robotics.net/robotics_league/upload/documents-2018/ERL_Consumer_10092018.pdf

[2] http://www.ycbbenchmarks.com/object-set/

[3] http://rgbd-dataset.cs.washington.edu/

[4] https://www.amazon.de/-/en/4939642-Berberil-N-Eye-Drops/dp/B00E5AN0NK/

[5] https://www.amazon.co.uk/Vizulize-Dry-Eyes-Drops-10ml/dp/B003BZVG2C/

[6] https://www.amazon.it/I-DEW-Aqual-Gel-Night-Time-lubrificante/dp/B00R7NX532/

[7] https://www.amazon.de/Livola-Tablettenbox-Pillendose-Regenbogen-Medikamentenbox/dp/B086S6J111/

[8] https://www.amazon.co.uk/AidShunn-Portable-Organizer-Supplements-Medication/dp/B088JZKVG3

[9] https://www.amazon.it/PortaPillole-Settimanale-Contenitore-Organizzatore-Promemoria/dp/B08GYC3M1R/

3. Depth image(s) from camera(s)

4. Pointcloud from 3D camera

5. Transformation tree (for extrinsic calibration of camera)

* indicates minimum requirement

4.3 Human Recognition

The list of target persons to be recognized must be selected at least 2 hours before the start of the trials. Teams can take pictures of the target persons at this time.

The topics to be recorded:

1. *RGB image(s) from camera(s)

2. *Intrinsic camera calibration

3. Depth image(s) from camera(s)

4. Pointcloud from 3D camera

5. Transformation tree (for extrinsic calibration of camera)

* indicates minimum requirement

4.4 Activity Recognition

The activities to be recognized are listed below. They are based on the activities in the ETRI-3DActivity dataset[10] and the Charades dataset[11].

ID	ETRI-3DActivity	Charades
1	Eating food with a fork	C156 Someone is eating something
2	Pouring water into a cup	C108 Pouring something into a cup/glass/bottle
3	Taking medicine	C129 Taking/consuming some medicine

[10] https://ai4robot.github.io/etri-activity3d-en/

[11] http://vuchallenge.org/charades.html

4	Drinking water	C106 Drinking from a cup/glass/bottle
5	Putting/taking food in/from the fridge	C142 Closing a refrigerator? or C143 Opening a refrigerator?
6	Trimming vegetables	
7	Peeling fruit	
8	Using a gas stove	
9	Cutting vegetable on the cutting board	
10	Brushing teeth	
11	Washing hands	C139 Washing their hands
12	Washing face	
13	Wiping face with a towel	
14	Putting on cosmetics	
15	Putting on lipstick	
16	Brushing hair	C144 Fixing their hair?
17	Blow drying hair	
18	Putting on a jacket	
19	Taking off a jacket	
20	Putting on/taking off shoes	C055 Putting on shoes or C057 Taking off some shoes
21	Putting on/taking off glasses	
22	Washing the dishes	C121 wash a dish/dishes
23	Vacuuming the floor	C137 Holding a vacuum?
24	Scrubbing the floor with a rag	C038 Washing something with a towel?
25	Wiping off the dining table	C013 Washing a table
26	Rubbing up furniture	C038 Washing something with a towel?
27	Spreading/folding bedding	C075 Tidying up a blanket(s)?
28	Washing a towel by hands	C037 Tidying up a towel? or C038 Washing something with a towel?
29	Hanging out laundry	C001 Putting clothes somewhere?
30	Looking around for something	
31	Using a remote control	
32	Reading a book	C032 Watching/Reading/Looking at a book
33	Reading a newspaper	

34	Handwriting	
35	Talking on the phone	C019 Talking on a phone/camera
36	Playing with a mobile phone	C016 Playing with a phone/camera
37	Using a computer	C052 Working/playing on a laptop or C051 Watching a laptop or something on a laptop
38	Smoking	
39	Clapping	
40	Rubbing face with hands	
41	Doing freehand exercise	
42	Doing neck roll exercise	
43	Massaging a shoulder oneself	
44	Taking a bow	
45	Talking to each other	
46	Handshaking	
47	Hugging each other	
48	Fighting each other	
49	Waving a hand	
50	Flapping a hand up and down (beckoning)	
51	Pointing with a finger	
52	Opening the door and walking in	C008 Opening a door
53	Fallen on the floor	
54	Sitting up / Standing up	C151 Someone is going from standing to sitting or C154 Someone is standing up from somewhere
55	Lying down	C122 Lying on a sofa/couch or C124 Lying on the floor or C134 Lying on a bed
56	*Limping	
57	*Colliding against furniture	
58	*Someone is coughing/sneezing	C153 Someone is sneezing

Table 2: List of activities for Activity Recognition

* not in ETRI-3DActivity
The topics to be recorded are:

1. *RGB image(s) from camera(s)
2. *Intrinsic camera calibration
3. Depth image(s) from camera(s)
4. Pointcloud from 3D camera
5. Transformation tree (for extrinsic calibration of camera)

* indicates minimum requirement

4.5 Gesture Recognition

The gestures to be recognized are listed below:

Gesture	Dataset(s)
Thumb Up	20BN-Jester[a], NDHGD[b]
Thumb Down	20BN-Jester
Stop Sign / Open Hand	20BN-Jester, NDHGD
Pulling Hand In / Call someone	20BN-Jester, NDHGD
Push Hand out / Pushing Hand away	20BN-Jester, NDHGD
Shake Hand	20BN-Jester, NDHGD
Pointing	
Nodding	
Shaking head	

[a] `https://20bn.com/datasets/jester`

[b] NVIDIA Dynamic Hand Gesture Dataset: `https://research.nvidia.com/publication/online-detection-and-classification-dynamic-hand-gestures-recurrent-3d-convolutiona`

Table 3: List of gestures for Gesture Recognition

The topics to be recorded:

1. *RGB image(s) from camera(s)
2. *Intrinsic camera calibration
3. Depth image(s) from camera(s)
4. Pointcloud from 3D camera

5. Transformation tree (for extrinsic calibration of camera)

* indicates minimum requirement

4.6 Handover

The topics to be recorded:

1. *RGB image(s) from camera(s)
2. *Intrinsic camera calibration
3. *Joint states of robot (including position, velocity, torques etc.)
4. *Gripper status/commands
5. Force-torque sensor
6. Tactile sensor
7. Depth image(s) from camera(s)
8. Pointcloud from 3D camera
9. Transformation tree (for extrinsic calibration of camera)

* indicates minimum requirement

4.7 Receive Object

The topics to be recorded:

1. *RGB image(s) from camera(s)
2. *Intrinsic camera calibration
3. *Joint states of robot (including position, velocity, torques etc.)
4. *Gripper status/commands
5. Force-torque sensor
6. Tactile sensor

7. Depth image(s) from camera(s)

8. Pointcloud from 3D camera

9. Transformation tree (for extrinsic calibration of camera)

* indicates minimum requirement

4.8 Relation to Cascade Campaigns

The datasets collected during the field campaign will be used for the cascade evaluation campaign. Due to the limited amount of data that can be collected, these datasets will be used as test/validation sets only, where possible. Therefore we list here datasets which can be used for training learning models, since at least a subset of the datasets overlap with the data collected during HEART-MET.

Task	Training dataset(s)
Object Detection	YCB[a], Washington RGB-D[b]
Human Recognition	IARPA Janus Benchmark[c] (Note: this dataset cannot be used for training, but can be used for developing algorithms since it evaluates a similar task)
Activity Recognition	ETRI-3DActivity[d], Charades[e]
Gesture Recognition	20BN-Jester[f], NVIDIA Dynamic Hand Gesture Dataset[g]
Handover/Receive Object	A multi-sensor dataset of human-human handover[h]

Table 4: List of potential training datasets

[a] http://www.ycbbenchmarks.com/object-set/

[b] http://rgbd-dataset.cs.washington.edu/

[c] https://www.nist.gov/programs-projects/face-challenges

[d] https://ai4robot.github.io/etri-activity3d-en/

[e] http://vuchallenge.org/charades.html

[f] https://20bn.com/datasets/jester

[g] https://research.nvidia.com/publication/online-detection-and-classification-dynamic-hand-ges

[h] Carf, Alessandro, et al. ”A multi-sensor dataset of human-human handover.” Data in brief 22 (2019): 109-117.