Lab1: Activity Detection

Release: Tue Jan 14

Due: Mon Jan 27, 11:59pm (IMU dataset), Jan 31, 11:59PM (doc, code and test result)

In this lab, you will learn how to connect to the robotic car, control the car through GUI and CLI (optional), retrieve data from the IMU unit. You will design algorithms to detect activities using the collected data.

This lab has two parts. Part I is data collection using the robotic car (and its IMU unit). This part requires at least 2 group members and 1 laptop (**you need to bring one**) per group. Part II is algorithm design for activity detection, which you will do offline.

Part I: You will need to collect IMU data traces for 4 types of activitie: you holding the car standing, you holding the car walking, you holding the car jumping, and car driving (using the GUI to control the speed). For each activity type, you will need to collect 2 IMU traces, each trace of at least 8 seconds in length. Each robot car has limited memory so do not go too long (< 60 seconds). You will need to submit the traces (marked by their true activity type) by 1/27, and we will also release all the traces on 1/28. **Traces that are not marked by their true activity type will NOT be accepted.**

Submission link: https://www.dropbox.com/request/mJBrAPSe6LB4TrMksv4l (due 1/27, 11:59PM)

Part II: Design algorithms for activity detection. You will need to submit a document which describes. your work and your code. On 1/28, we will also release a test dataset that you will run your algorithm and submit the result (standing, walking, jumping, driving (also show speed)).

Submission link: https://www.dropbox.com/request/vL9NKYSFGhebiXjuWwzs (due 1/31, 11:59PM)

Grading: 20% on submitting the IMU dataset, 40% on algorithm documentation and code, 40% on accuracy of result from the test dataset.

For those using **Windows**: this instruction was originally written for unix/linux OS, so some commands may not be available on your machine. Install *Cygwin* to run those commands on your Windows machine. For more information about *Cygwin*, refer to https://cygwin.com .

Step 1: Connect to the robotic car

This lab requires you to connect to the robot car (to retrieve the IMU data and to control its driving). The robotic car assigned to you will serve as an WiFi access point (AP) with name like "Pi_AP-1c54", join the AP just as it is a WiFi AP. Please refer to the time slot assignment page for the exact AP name of the robotic car assigned to your group. Note that once you are connected to the robot AP, you will not be able to connect to the Internet.

Step 2: "SSH" to the car to run commands on it

After you connect to the robotic car, use `ssh` to run commands on the robotic car:

open a terminal on your laptop, type (`$` just indicates the beginning of a command, do not type it in):

```
$ ssh stu@192.168.42.1
```

after you enter the password (listed on the assignment page), you will get full access to the car, and you can now on run commands on the car just like your laptop.

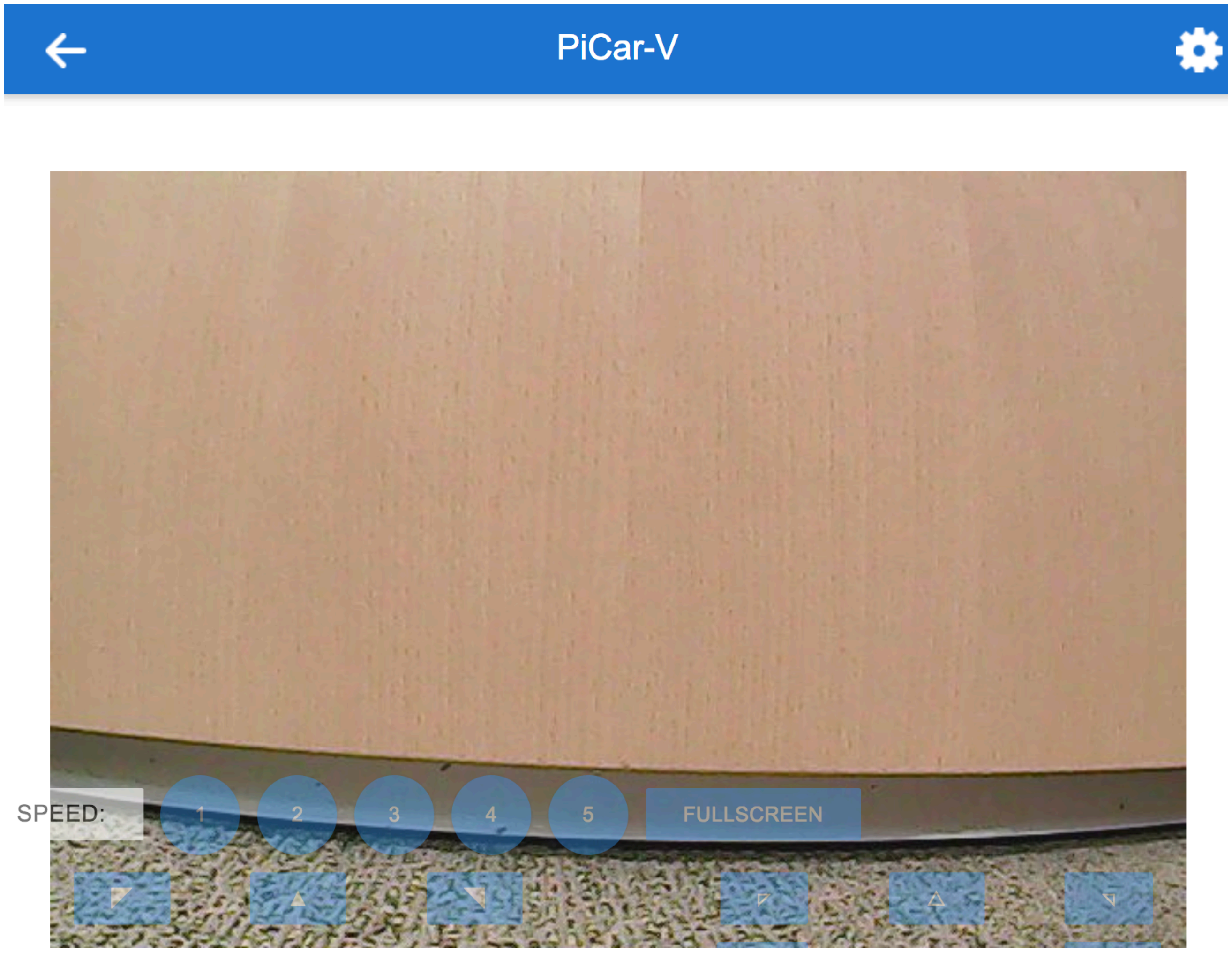
Step 3: Check the working environment

We have created a working directory, `~/teamX` (substitutes X for your team number, the same below), for you. `record.py` is now in that folder. Use `cd ~/teamX` to go to that folder and `ls ~/teamX` to see what are inside the folder.

Remember to save all the files you need and all data you create to that folder, and copy back to your laptop at the end of your time slot. (See Step 6 about transfering files)

Step 4: Drive the robotic car through web interface

After connecting to the robotic car, go to http://192.168.42.1:8000 on your laptop, you will see something like this:



- Press *'w'*, *'s'*, *'a'*, *'d'* on your keyboard to move forward, move backward, turn left and turn right respectively.
- Press *arrow up, arrow down, arrow left, arrow right* to turn the camera up, down, left and right respectively.-
- You can adjust speed by clicking on the buttons located at bottom left.
- Ignore other buttons on the web page, they are not working now.

***Note: drive with caution. We only have 3 robot cars and they are delicate (especially since it has a camera in front of the car). If you damage it, then the teams after you will not have any car to do experiments, and you will not have a car for lab 2 and 3. So Drive SLOWLY.* The TAs have the authority to remove you from the experiments at any time.**

Step 5: Create your dataset

When collecting IMU measurements, two team members will participate. One member will hold the car to perform activity, another will record the data trace on the laptop.

To record the activity, under your working directory, run `python record.py` to record one data trace, for example:

```
$ python record.py --team=0 --type=Walking --data-number=0 --length=10
```

By running this command, you will get a file named `activity-team0-Walking-0.txt` which contains 10 seconds of IMU data. Remeber to modify parameters of this command before each time you are going to record. **Appendix** provides the format of the collected IMU data.

You should record at least 2 data traces for each of the 4 activity types (`Walking`, `Jumping`, `Standing` and `Driving`), each data trace should be at least 10 seconds but no more than 60 seconds.

After you recorded all the data segements you need, run following command to compress them into one file:

```
$ tar -cvf activity-dataset-teamX.tar activity*.txt
```

Step 6: Transfer the dataset to your laptop

While connected to the robotic car, use `scp` to transfer files between your laptop and the robotic car.

Open a terminal on your laptop, run:

```
$ scp stu@192.168.42.1:~/teamX/activity-dataset-teamX.tar DESTINATION
```

Substitues `DESTINATION` for path of a folder on your laptop.

A full example:

```
$ scp stu@192.168.42.1:~/team0/activity-dataset-team0.tar ~/Downloads
```

After running this command, `activity-dataset-team0.tar` should appear in `~/Downloads` on your laptop. **Double check your laptop before you leave the lab.**

Step 7: Submit your dataset

**Upload your dataset (marked by your group name) to the following dropbox file

Submission link: https://www.dropbox.com/request/mJBrAPSe6LB4TrMksv4l (due 1/27, 11:59PM)

We will collect datasets from all the teams to form a big training dataset and put it on Piazza, so you can have more data to play with.

Step 8: Detect activity

With your dataset (and the class-wide dataset), you will need to extract the accelerometer values (see the data format in the Appendix) and use these values for activity detection. Write a document explaining your algorithm design.

Submission link: https://www.dropbox.com/request/vL9NKYSFGhebiXjuWwzs (due 1/31, 11:59PM)

You can refer to [Cence2008]Sensing Meetings Mobile Social Networks: The Design, Implementation and Evaluation of the CenceMe Application, Sensys08 for one possible method. Other methods are also accepted.

Appendix

Drive the car via Python

The library to control the car is located at `/home/pi/SunFounder_PiCar/picar`, to use it, link that folder to your working directory by running `ln -s /home/pi/SunFounder_PiCar/picar /home/stu/teamX` (substitute X for your team number).

Then in your python code, import these two file and instantiate the two classes:

```
from picar.front_wheels import Front_Wheels
from picar.back_wheels import Back_Wheels

front_wheels = Front_Wheels()
back_wheels = Back_Wheels()
```

There are four functions allow you to turn front wheels: `turn_straight()` turns the front wheels back straight; `turn_left()`, `turn_right()` turns the front wheels left and right respectively; you can also let front wheels turn to a specific angle by `turn(THE_ANGLE)`, $70 \leq \text{THE\_ANGLE} \leq 110$.

There are three funtions allow you to control back wheels: `forward()` moves both back wheels forward, `backward()` moves both wheels backward, `stop()` stops both wheels. Once you invoke `forward()` or `backward()`, the wheels will keep moving until you invoke `stop()`. To control the speed set `speed` between 0 to 100, 100 means the maximum speed. You are able to adjust the speed while the wheels are moving.

A full example controls the car to make a 'three point turn':

```python
from picar.front_wheels import Front_Wheels
from picar.back_wheels import Back_Wheels
import time

front_wheels = Front_Wheels()
back_wheels = Back_Wheels()

try:
    # set front_wheels back to straight
    front_wheels.turn_straight()

    # first part
    # set the speed to 50
    back_wheels.speed = 50
    front_wheels.turn_left()
    back_wheels.forward()
    time.sleep(4)
    back_wheels.stop()

    # seconde part
    # set the speed to 50
    back_wheels.speed = 50
    front_wheels.turn_right()
    back_wheels.backward()
    time.sleep(4)
    back_wheels.stop()

    # third part
    back_wheels.forward()
    for i in range(70, 90, 1):
        back_wheels.speed = (90 - i) * 4
        front_wheels.turn(i)
        time.sleep(3. / 20)
    back_wheels.stop()

except:
    print "encountered error or interrupted, motor stop, turn straight"
    back_wheels.stop()
```

Details on Reading Data from the Car IMU

We have provided "record.py" so that you can automatically record the IMU sensor readings for each activity. To explain how we build "record.py", the following shows how one can obtain sensor data from the IMU on the robot car.

The library to get data from IMU is provided at `/home/pi/IMU`. Link this file to your working directory by `ln -s /home/pi/IMU /home/stu/teamX`.

Then, in your python code, import the class: `from IMU import IMU`.

Use `read()` to get data. The return value of this function is a *dict* like:

```
{'xGyro': -44, 'yGyro': 29, 'zGyro': -18, 'xMag': -884, 'yMag': -61, 'zMag': 327,
'xAccl': -48, 'yAccl': 16, 'zAccl': 1386}
```

A full example:

```
from IMU import IMU

imu = IMU()
for i in range(0, 10):
    print imu.read()
```

Transfer files to the car

Open a terminal on your laptop, run:

```
$ scp SOURCE stu@192.168.42.1:DESTINATION
$ scp -r SOURCE stu@192.168.42.1:DESTINATION
```

Remember to substitue the real path for the capitalized part.

Example1: copy the file "foobar.py" from home directory on your laptop to the car. Start a terminal on your laptop, run `scp ~/foobar.py stu@192.168.42.1:~/team1` on your laptop, and you should see "foobar.py" under `~/team1` on the car.

Example2: copy the directory "foobar" from home directory on your laptop to the car. Start a terminal on your laptop, run `scp -r ~/foobar stu@192.168.42.1:~/team1` on your laptop, and you should see the directory "foobar" under `~/team1` on the car.

Activity data segment format

Each data segment includes `type` : type of the activity; `seq` : data sequence which is a *list*, and each element of the list is a *dict* contains `time` and `data`. `time` stores the value returned by `time.time()`, `data` stores the value returned by `IMU.read()`.

A full example with sequence length 2:

```
{'type': 'Walking',
 'seq': [{'time': 1515728861.742783, 'data': {'xGyro': -44, 'yGyro': 29, 'zGyro':
-18, 'xMag': -884, 'yMag': -61, 'zMag': 327, 'xAccl': -48, 'yAccl': 16, 'zAccl':
1386}}, {'time': 1515728862.742783, 'data': {'xGyro': -46, 'yGyro': 27, 'zGyro':
-16, 'xMag': -880, 'yMag': -60, 'zMag': 328, 'xAccl': -49, 'yAccl': 17, 'zAccl':
1385}}]}
```