Version: January 17, 2022

© Computer Vision Laboratory

Linköping University

Image Processing in Python
Tutorial for TSBB15

1 Introduction

During this exercise, the goal is to become familiar with Python and the NumPy library. You should also get a better feeling for how images are represented as matrices as well as the connection between mathematical expressions and the Python code to implement them.

We additionally provide a guide for how to get started with using an IDE for python, which will make your life much easier during the projects.

2 Exercises

1. Make sure that you've downloaded the notebook for this lesson, either at

    https://gitlab.liu.se/cvl/tsbb15 or

    https://www.cvl.isy.liu.se/en/education/undergraduate/tsbb15/

2. Now, if you're using the school computers, e.g. olympen, open a terminal and write the following commands.

    ```
    $ module load prog/opencv/3.4.3
    $ module load courses/TSBB15
    $ jupyter notebook --NotebookApp.iopub_data_rate_limit=1.0e10
    ```

    The first line includes the OpenCV module, which is a library of computer vision related operations that is used in this course. The second line activates the parts specific to this course. The third line starts a jupyter notebook. Find the correct notebook and open it. Follow the instructions in there.

3 Good To Know Things

This section covers unit tests, and development tools. None of this is strictly part of the course, but they are very useful.

3.1 Development Tools

There exist a large number of different development tools for python, ranging from a text-editor like sublime text that provides minimal help, but also makes it easy to get started, to full IDEs, such as `pycharm`. Getting started using an IDE such as pycharm might require some time to get familiar with the program, but it will usually save time in the long run due to the very useful auto-complete functions and integrated debugging functionality.

For the projects we strongly recommend that you use `pycharm` or `vscode`. They can be added by

```
$ module load prog/pycharm-professional/2021.2.3
$ module load prog/vscode/1.52
```

Once you have loaded the modules you can start them with:

```
$ code &
$ pycharm.sh &
```

Task 1: Open vscode. Try creating a python file and running it.
Task 2: Open pycharm. Try creating a python file and running it.
Both these IDEs will provide you with a good development environment. Choose whichever you feel more comfortable with.

3.2 Unit tests

When writing programs larger than a few lines it is often useful to test the intermediate steps of the program in isolation. There are (at least) two reasons for this: It helps you define what each part of the program should actually do, and it makes sure that it actually does what you intended in isolation from the other parts. For larger projects involving multiple people unit tests can allow you to test each component in isolation for some simple cases.

A unit-test is in practice a nothing more than a function that calls the function you want to test with arguments that should produce some known output, for example:

```python
def test_addition():
    a = 10
    b = 15

    c = 25

    assert a + b == c
```

This function tests that the addition operation behaves as expected. Often you would write multiple tests so that you can be sure that edge cases are handled correctly. A more complicated example for a function that projects a point in an image:

```python
def project_point_in_image(point, camera_matrix):
    image_point = camera_matrix @ point

    return image_point
```

With the tests being:

```python
def project_camera_position():
    #Project a point that is at the same position as the camera into the image
    pass

def project_behind_camera():
    #Project a point that is behind the camera into the image
    pass

def project_point():
    #Project a sensible point that is in front of the camera into the image
    pass
```

Try to implement the missing test cases. It is recommended to use the same camera matrix in all tests, but different points. Note that the point of this exercise is as much to implement the tests as it is to think about what the result should be in each case.