Guidelines for Programming Assignments
Computer Vision CS 766, Spring 2024

This document aims to clarify important issues relevant to programming assignments for the course. Following the guidelines explained here will greatly simplify submission, testing, and grading of your work.

Academic Honesty Policy

The course requires you to read the Code of Academic Integrity [1] as stated by University of Wisconsin-Madison.

You are permitted and encouraged to discuss ideas with other students. However, you may not share answers or your code with others. Also, copying code/solutions from other sources, including online files, PYTHON sources or solutions from other educational institutions, but not limited to these is prohibited. We will be monitoring source code for individuality. Ramifications for students who violate the Code of Academic Integrity may include an "F" grade for the course, documentation on your academic file and even probation.

Software

Programming Languages
The official programming language for this course is PYTHON. The best resource to learn PYTHON is the official documentation [2], which is an excellent database of commands and examples. There are several ways to access the documentation:

- **From the web.** The entire documentation is available online.
- **From within Python:** You can access the documentation directly from your Python environment by using the built-in help function. Simply type `help(function_name)` to display the documentation for a specific function or module.

If you are new to PYTHON, the official Getting Started Guide and tutorials [3] are good resources to get you started.

Special Notes on Using Built-in PYTHON Functions
PYTHON comes with a large collection of functions and toolboxes. You will use PIL, numpy and matplotlib packages for the basic image operation and visualization.

In the context of this course, some of the classic algorithms have been implemented. **Thus, only basic functions are allowed in assignments, unless otherwise specified.**

In addition, the usage of one or more **specific commands may be barred**. Special instructions regarding allowed or disallowed functions will be explicitly stated in the description of each programming problem.

Pay attention to the instructions specified in each assignment. If you are unsure whether a built-in function can be used or not, please ask before assuming! You will receive no credit for using built-in functions where you should be coding them. When in doubt, ASK.

Working with Images in PYTHON

- **Image memory representation.** An image is represented as a matrix in PYTHON. In general, a grayscale image is represented as an m (rows) x n (columns) matrix, where m corresponds to the height of the image in number of pixels, and n corresponds to the width of the image in number of pixels. Similarly, a color (RGB) image is represented as an m (rows) x n (columns) x 3 matrix, where the three "slices" of the matrix represent the red, green and blue channels of an image.
- **Image coordinate system.** As mentioned above, an image is represented as a matrix in PYTHON. Therefore, we can retrieve the intensity of a pixel in the same way we retrieve an element of a matrix. Suppose we want to retrieve the intensity of a pixel P at location x along the X-axis and y along the Y-axis of an image A, it is conventional in literature and in our course notes to write as `P=A[x,y]`. However, the ordering of pixel indices is reversed in numpy.ndarray. **To retrieve the intensity of pixel P in the XY coordinate system, the correct numpy expression is `A(y,x)`, not `A(x,y)`!** This is often a source of confusion (and may be a source of bugs in your code).

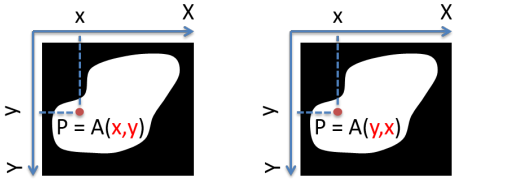

Image A in Literatures and Lecture Notes

Image A in numpy.ndarray

Comparison of Pixel Index Ordering in Lecture Notes and in numpy.ndarray

Running and Testing

For each homework, a file named **runHwX.py** will be provided (where X indicates the homework number). This file will serve as a framework for the homework and guide you through the individual programming milestones. In some cases, you will need to edit this file to complete the skeleton code or fill in certain parameters. It also serves as the interface to all functions/programs you need to implement and will help you run and test your solution. Therefore, your functions must strictly adhere to the specified signatures.

runHwX.py contains a set of sub-functions prefixed with the string "walkthrough" or "challenge," which are test cases for walkthroughs and challenges. Make sure you can run **runHwX.py** without errors before submission.

Submission

The homework should be submitted electronically **before midnight** on the due date. Submission should contain:

- **PYTHON** source files
- **runHwX.py**. This file is provided in each assignment. Make sure to fill in your own name, NetID, filenames, parameters, etc.
- **README**. In this file you should briefly explain your design decisions, the parameters/constants you used in the algorithms and describe any features that you have incorporated into your programs in addition to what was required.
- **Output files,** if requested in the homework problem document.

Include all the files in a zip file named **hwX_yourNetID.zip** (where X is the homework number) and upload to Canvas.

References

[1] University of Wisconsin-Madison Academic Integrity. [Online]. https://conduct.students.wisc.edu/academic-misconduct/

[2] PYTHON Documentation . [Online]. https://www.python.org/doc/

[3] PYTHON for Beginners. [Online]. https://www.python.org/about/gettingstarted/