Computational Photography HW0

Contents:

1. Introduction

In this first homework we will learn about how to read and write images, how images are represented in python, and how we can manipulate them. Before starting this homework, you need to install python and the other packages. This process is outlined in the installation instructions, accessible on the left bar of the coursera class page.

For an experienced python user, this homework assignment should only take a few minutes. For people with only a little programming experience, this assignment should take about 1-2 hours.

2. Working with images

In this homework, we will be learning a bit about the tools we just installed. We will read and write images, and play around with the internal representation of an image. Please refer to the coding video tutorials, and additional references at the bottom of this document to help you find your way.

a. Splitting an image into rgb channels

In the HW0 folder for this assignment, you will find a file part0.py.

We are going to implement the split_rgb function. You will find the specification for the function and a description of what it needs to do in the file. Please make sure to read the specification carefully, and that your implementation matches it exactly.

At the bottom of the file, you will see the ‘main’ and ‘test’ functions, which will apply unit tests to your code, load images from the /images/part0 subfolder, apply your function to them, and save

the outputs (if any). These functions will be called when you run the part0 file, so you can use it to make sure your function works correctly. When you are finished, you can run the submit.py script in the hw0 folder to get credit for completing the assignment.

This function can be implemented in a few lines of code using slice notation. For more details on slice notation, see the references section at the bottom of this document.

b. Reconstructing an image

In the HW0 folder you will find a file part1.py.

We are going to implement the interlace function. You will find the definition statement for the function and a description of what it needs to do in the file. Same as above, this file also provides some handy testing code which will run unit tests on your function, and then apply it to the two images in the images/part1 folder.

When you are ready, submit.py will grade your function. You might find the numpy.arange function useful while implementing this part of the assignment. Consult the numpy documentation in the reference section below.

c. Converting to grayscale

The code for this assignment is cointained in part2.py

Your task is to implement the grayscale function, which converts a color image into a grayscale image. The doc string for the function contains the details. Again, there is test code included in the file, which will look through the images/part2 folder and save grayscale versions of all files found there.

You may find the functions numpy.sum and numpy.astype useful while implementing this part of the assignment. Consult the numpy documentation in the references section below.

3. Suggested reading

numpy array attributes and methods - link
indexing and slicing numpy arrays - link
reading images in opencv (take note of the color order at the bottom) - link
numpy user guide - link
numpy and scipy documentation - link
opencv documentation - link