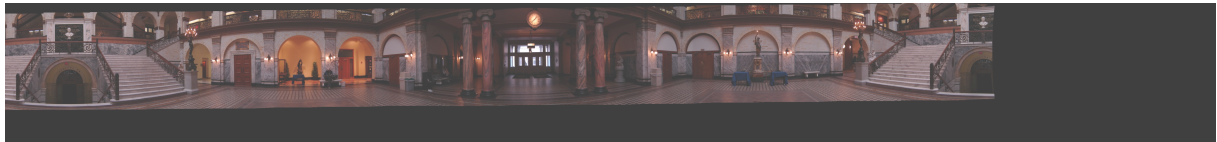


Homework 3

Costa Huang (sh3397@drexel.edu)

For this assignment, since we didn't have to run it by using different parameters, there isn't much to show in the result section. I got (almost) the same result as the example outputs off by 1 pixel... My image shape is (528, 4903) while the example provided has the shape (527, 4902). I tried to round down to the same dimension, but the code would just break. Not exactly sure why, but I figure it is probably related to the fact that my `pyramid_lucas_kanade()` is off by `0.0052080` compared to the test case result.

Anyway, while I was trying to figure out the proper height and width for the image, I tried with shape (600, 5000) and it looks like the following. It gave me the confirmation that I was going to the right direction.



Then I figured out how to use the final displacement and the image shape and it produces the following image by running the command below.

```
$ python3 hw3.py example_inputs/files.txt example_output1.png
```



Struggles

I spent a lot of time debugging `lucas_kanade()`. The instructions weren't exactly clear. For example, it says "use a *normalized* 3x3 sobel kernel", but normal matrix means something like

A is **normal** if it [commutes](#) with its [conjugate transpose](#) A^*

I wasn't able to get it working until I received email response from Professor Kratz.

Also the way mask works `I_x * I_x * mask` is also kind of a magic. Sometimes I feel like programming in python is to find magic that works but may prohibit understandings.

Lastly, I struggled with `pyramid_lucas_kanade()`, which has code

```
disp = initial_d / 2.**levels
for level in range(levels):
```

But the problem is when you build the gaussian pyramids, level 0 means the original image, which is the opposite of the coarsest level. The prepopulated code should have been:

```
disp = initial_d / 2.**levels
for level in reversed(range(levels)):
```

I feel slightly frustrated by this because it wasn't detected by the test case `test_gaussian_pyramid`, which only checks the shapes. Perhaps the instructions could specify the direction of coarseness so we know what to expect.

Written with [StackEdit](#).