CS 4475-RBA - Computational Photography Summer 2015

Project #2 - Due Friday June 26th, before 11:54pm

Part 1: Description

You have 2 remaining opportunities to use what you have learned to create a computational photography artifact. The main goal is to explore an aspect of computational photography that excites you, using/reinforcing what you have learned in class, and hopefully have fun! You can also use this as an opportunity to build up a portfolio of work that may be interesting to potential employers or research advisors.

In the past students have build systems to do motion blurring, image mosaics, image morphing, colorization, image blending, producing cinemagraphs, image seam-carving, hole- filling, simulating tilt-shift, capturing images to get a different output from photosynth, simulating how a picture would appear rendered on wood, steel, etched papers (etc.), generating non-photorealistic / painterly / sketch images, photo mosaics, photo montages, photo collages, averaging and aligning faces, etc. Feel free to share more ideas over Piazza.

There has to be a computational aspect to your project. That is, you can't just take a bunch of images and produce one artifact. It has to be a repeatable process that is applicable to a different set of input images.

Choice of tools:

You are required to use Python/OpenCV as your main tool in at least two of the three projects. (You may use it for all three.)

Part 2: Assignment Deliverables

You will need to turn in:

- A project writeup as a PDF file. (see more details below)
- Any source images you used.
- The source code that you ran. (if any)
- An example output (image/images/video) that your workflow produced.
- An HTML (.html) file suitable for display on the class website that highlights your project deliverable. (See the attached template and description below.)

The writeup is the most important document, followed closely by the html file and example output.

Your **writeup** should be a PDF file that has the following information:

1. Your name(s), email address(es), and T-Square ID.

2. A high level description of your project (what it does / is / the purpose)

The images showing example input/output should be embedded into your document in this high level description so that we do not have to open the source/output images you submit in addition to the PDF file while reviewing the PDF file.

3. A lower level description of your “workflow”. You must specify your inputs (e.g. “Two images of the same size”), each step in your computation/procedure (e.g. “Step 1, the cropImages function makes sure that both images are the same size.”), and end with the output (e.g. “The final step concatenates the 5 images together into an animated gif.”)

   We suggest that you draw a flow chart or diagram graphically outlining your process, combined with textual details of each step. If your project makes use of Python/OpenCV code, you may simply refer to each function in your code. If your process is manually operation of a tool, you must specify enough detail so that another person could replicate that “box” in your flowchart or workflow diagram.

   Again, the description of your workflow should be at a level such that somebody with all of your submitted files could use different input images to produce similar results.

4. Any code or ideas you did not directly develop must be credited with their source (another student in the class, a website URL, etc...)

5. Finally, you must specify how you would like the “Artistic” / “Technical” grading breakdown to apply to your project. Each can be between 10 to 50 points (the total must add up to 60 points). If you do not specify, we will apply a 30/30 breakdown. Valid options are:

Artistic: 15	Technical 45	Artistic: 45	Technical 15
Artistic: 20	Technical 40	Artistic: 40	Technical 20
Artistic: 25	Technical 35	Artistic: 35	Technical 25
Artistic: 30	Technical 30	Artistic: 30	Technical 30

Your **HTML** file should have the following attributes:

1. Your name (if you want to be publicly identified with the work), or a pseudonym (if you would like to protect your privacy).
2. A brief blurb describing the motivation/purpose/procedure for your project.
3. An example output image/video/animation.
4. *Optional contents: You may, but are not required to, include links to any document you have uploaded as part of your t-square submission, such as your source code (to replicate the work) or alternate input/output examples.*

Please make sure that all links are “relative” and refer to only the file names you use in your t-square submission. The goal is that if we place the HTML file and your associated upload files (.jpg / .py, etc...) in a directory, they will be complete.

We may choose to show only a subset of the submissions on the class website from each project. The expectation is that every student will have a submission for one of the three projects that is suitable for exhibition.

Grading Rubric:

- Choice of input Images (15%) - Was your choice of input images appropriate to fully demonstrate your process/pipeline? Were they visually interesting? Do you have the appropriate rights/permissions to display the output publicly? (e.g. do you own copyright yourself, or were they creative commons licensed or in the public domain?)

- Writeup / Report / HTML file (25%) - Does your submission include all required elements in the correct format? Is there enough detail that a reader could replicate your pipeline/workflow?

- Artistic and Technical aspects (60% overall) you choose the point breakdown:
  - Artistic (15-45 points) - A mostly subjective measure of how “Awesome” the output of your project looks (irregardless of how technically challenging it was to produce). Was the effect novel and interesting? Is it clearly “artistic”? (whatever that means...) Is it visually pleasing? Does it have a consistent message or theme?

  - Technical (15-45 points) - How much your project demonstrates that you have learned about computational photography, including but not limited to: Python / OpenCV or third party software tools. Did you learn novel API calls, tools, etc that were not covered in class or did you simply re-use code we provided? Does your workflow/process follow the most efficient path, or could it be simplified? Is it subjectively “Awesome” due to the technical sophistication?