

Due Date: 23:00 pm on Friday, November 2nd, 2018

Photoshop Battle

In this assignment, you are going to implement some basic operations to simulate Photoshop techniques and try to give the best results among your colleagues. You are going to use images that are given from MSRA10K [1] dataset which have foreground object masks as well. First, you will separate the foreground object by using the given mask and then put it on a different image to change the background of the foreground object. You will set the position of the foreground object by changing the location of the pixels in order to put it on a meaningful place. You may crop the foreground object/image and its mask accordingly. After that, you will add some depth to your image by blurring the background and sharpening the foreground.

At the end, as a bonus point, you may smooth the edges of the foreground object to decrease the Photoshop effect and get more realistic results. Each experiment you did will be graded accordingly, even it has failed.

Note: You will use the given example background image for “im1”, but for the other images, use your imagination and find proper backgrounds to Photoshop and be the best among others!



Figure 1: Example results.

PART 1

Image Masking

You are expected to use related image masks to extract foreground objects. Basically, you are going to consider only the pixel values which is on the location stated in image mask with “255”s. After obtaining foreground object’s pixel values, you are going to save the new image with a black background. The steps of this operation can be seen in below formula,

$$F(x, y) = \begin{cases} 0 & \text{if } M(x, y) = 0 \\ I(x, y) & \text{otherwise} \end{cases}$$

where F is the foreground image, I is the original image, M is the mask and (x,y) denotes the coordinate of a pixel value in an image.

Changing Background

Now you are going to overlay the foreground image over the given background image. The simplest way to do this is to apply the given formula,

$$P(x, y) = \begin{cases} B(x, y) & \text{if } F(x, y) = 0 \\ F(x, y) & \text{otherwise} \end{cases}$$

where P is the Photoshopped image, B is the background image and F is the foreground image. You may crop (see Figure 1 (c)) and translate (see Figure 1 (d)) foreground object's pixels in order to put it to the proper place on the background image.

PART 2

Background Blurring

You are expected to apply Gaussian Blur only to the background image by implementing your own filter function. Basically, you will convolve the 2D Gaussian kernel with only the background image and apply it each channel of the color image. You may use built-in functions for the convolution but you should create your own filter kernel and you cannot use built-in Gaussian filter.

As a suggestion, you can create 1D Gaussian first and then make it a 2D kernel by matrix multiplication. In order to apply your filter to the entire image, you should consider one of the extrapolation methods to fill the boundaries such as copy edge, reflect across edge etc.

Your Gaussian filter function should automatically determine a reasonable kernel size for the filter. Give experimental results over different kernel sizes and sigma values for this blurring operation with your detailed comments and explanations.

Sharpening Foreground

Now you are going to implement an Unsharpen filter: 1) smooth the foreground image with Gaussian filter 2) take the difference from the original image to get high pass values 3) do alpha blending with this difference image on original image (basically $unsharpen_image = alpha(diff_image) + original_image$).

The Unsharpen filter function should take two parameters: Standard deviation of Gaussian (sigma) and blend weight (alpha). You are going to give experimental results with different alpha values to show the effect of it.

Now you have a blurred background and sharpened foreground. You will apply the same steps in Part-1 such as masking and overlaying to see the difference (we added an effect like portrait mode).

Smoothing the Edges (Bonus)

Since you have basically cut and pasted the foreground object onto a different background, now it is time to decrease this Photoshop effect in order to get realistic results. In this part, you may apply Savitzky-Golay filter [2] to the edges of the foreground object only. For this purpose, you should find the coordinates of the foreground object. You may think the foreground object as a blob from its mask and find the contours using some library functions. Then you can apply Savitzky-Golay filter by using built-in functions. Other solutions are welcomed as well.

The Implementation Details

1. You are not allowed to use built-in functions for image masking, overlaying and implementing filters. You may use convolution functions from libraries.
2. You are expected to write functions for each operation and give a main script to run all results.
3. You should pay attention to code readability such as comments, function/variable names and your code quality: 1) no hard-coding 2) no repeated code 3) cleanly separate and organize your code 4) use consistent style, indentation 5) write deterministic algorithms
4. Your code should read all images from a folder named "images" and save all the results in folders as "part1" and "part2".
5. Implement your code with Python 3 and use libraries from Anaconda. You can install any library that is not in Anaconda as well, such as OpenCV 3.

What should you write in the report?

- Give your resulting images step by step for each part.
- Give experimental results, used parameters and comments on the results in detail.
- Explain your implementation and code as well as your approach to the problem.
- A basic structure might be: 1) Introduction (what is the problem, how do you approach to this problem, what is the content of your report) 2) Implementation Details (the method you followed, the organization of your code, functions and details of your solution) 3) Experimental Results (all results for separate parts with different parameters and your comments on the results) 4) Conclusion (what are the results and what are the weaknesses of your implementation, in which parts you have failed and why, possible future solutions)
- You should write your report in \LaTeX

Grading

The assignment will be graded out of 100:

- **30 (part 1):** CODE: 0 (no implementation), 5 (a partially correct solution), 15 (a correct solution) and REPORT: 15
- **70 (part 2):** CODE: 0 (no implementation), 25 (a partially correct solution), 40 (a correct solution) and REPORT: 30
- **25 (bonus):** CODE: 0 (no implementation), 10 (a partially correct solution), 20 (a correct solution) and REPORT: 5

Late Policy

You have three days for late submission. You will lose 10 points from maximum evaluation score for each day (your submitted study will be evaluated over 90, 80 and 70 for each late submission day). You have to submit your solution in deadline date + three days, otherwise it will not be evaluated.

Academic Integrity

All work on assignments must be done individually unless stated otherwise. You are encouraged to discuss with your classmates about the given assignments, but these discussions should be carried out in an abstract way. That is, discussions related to a particular solution to a specific problem (either in actual code or in the pseudocode) will not be tolerated. In short, turning in someone else's work, in whole or in part, as your own will be considered as a violation of academic integrity. Please note that the former condition also holds for the material found on the web as everything on the web has been written by someone else.

References

- [1] <https://mmcheng.net/msra10k/>
- [2] Gowri, B. Ganga, et al. "2D Image data approximation using Savitzky Golay filter—Smoothing and differencing." Automation, Computing, Communication, Control and Compressed Sensing (iMac4s), 2013 International Multi-Conference on. IEEE, 2013.