CMPT 732: Practices in visual computing I
Assignment 1 - part 2
Image reconstruction + Poisson blending

Total points: 20 points

Due: Wednesday, 12 October, 3 PM

Part 1: Image reconstruction (10 pts)

In class we saw how we can reconstruct an image from only the first order derivatives, by solving a least square problem. In this part of the assignment you are going to implement image reconstruction using **second order** derivatives. We denote the source image as S and the target image as V. Additionally, we denote the pixel intensity at location (x, y) in S and V as $s(x, y)$ and $v(x, y)$ respectively. For each pixel (x, y) We can write:

$$V_{up}(x,y)+V_{down}(x,y)+V_(left)(x,y)+V_{right}(x,y) = S_{up}(x,y)+S_{down}(x,y)+S_(left)(x,y)+S_{right}(x,y), \quad (1)$$

where $V_up(x, y) = v(x, y+1)-v(x, y)$, $V_down(x, y) = v(x, y)-v(x, y-1)$, $V_left(x, y) = v(x, y)-v(x-1, y)$, and $V_right(x, y) = v(x + 1, y) - v(x, y)$, are the directional gradients of the target image. We can similarly define S_{up}, S_{down}, S_{left}, and S_{right}. We can re-write the above equation as:

$$4v(x,y)-v(x-1,y)-v(x+1,y)-v(x,y-1)-v(x,y+1) = S_{up}(x,y)+S_{down}(x,y)+S_(left)(x,y)+S_{right}(x,y). \quad (2)$$

This way, we will have a linear equation for each pixel, and similar to what we had in class, we can form a list square optimization and solve for the values of V. More specifically, by writing the coefficients of the above equation in a matrix A, we can write the set of equations for every pixel as $Av = b$. If the image has k pixels, A will be a $K \times K$ matrix, and v and b will be $k \times 1$ vectors.

Note that at the edges, for example at position $(0, 10)$, there is no left neighbor. Therefore you should not compute horizontal second derivatives. i.e, the equation becomes $V_{up}(x, y) + V_{down}(x, y) = S_{up}(x, y) + S_{down}(x, y)$. Also in the four corners, the equations reduce to $0 = 0$. So, in order for A to be full rank and the equation have a unique answer, we should impose four extra constraints. You should impose these

constraints on the intensities of the four constraints. i.e., $v(0,0) = c_1$, $v(0,n-1) = c_2$, $v(n-1,0) = c_3$, and $v(n-1,n-1) = c_4$, where c_is are constant values.
To implement this part of the assignment, please complete the `reconstruction.py` file. In your report you should visualize the reconstruction results, and also compute the least square error $|Av - b|$.

Part 2: Poisson blending

In this section you will implement Poisson blending. The framework provided, lets you:

1. select a patch from the source image
2. select the part of the target image where you want to paste the patch
3. rotate and/or scale the patch

In order to start the process, you should first select a patch from the source image by clicking on the points of the patch polygon on the image window which will open when you run the `main.py` function. after you select the patch, press any key to go to the next step. In the next step you should select the position of the patch in the target image. You can rotate the patch by pressing **r**, scale down the patch by pressing **a**, scale up the patch by pressing **s**, or move the patch to any direction by pressing the **arrow keys**. After you have selected the position of the patch, press **q** to go to the next step.
In the next step, you should complete the function `poisson_blend` in the `main.py` file.
In your report, please visualize multiple results with the source/target pairs that are provided. You should also report the least square error $|Av - b|$ for all of your experiments.
Note that it is better to select the patch and the pasting regions in the feature less areas of the both source and target images.