EE 569: Homework #3

Issued: 02/13/19 Due: 03/03/19

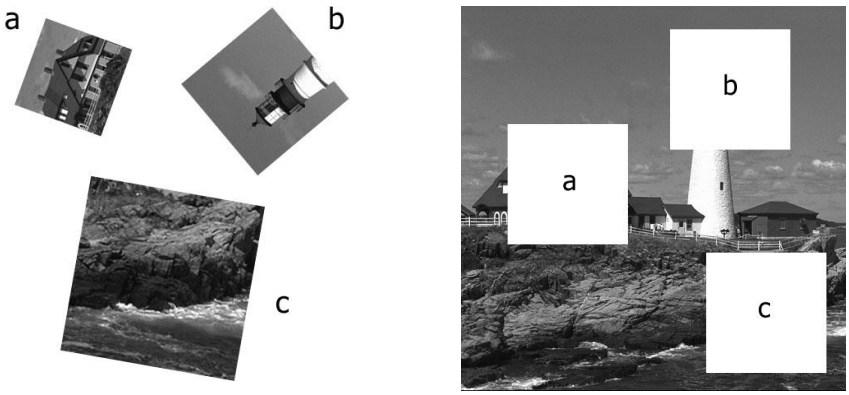
General Instructions:

1. Read *Homework Guidelines* and *MATLAB Function Guidelines* for the information about homework programming, write-up and submission. If you make any assumptions about a problem, please clearly state them in your report.
2. You need to understand the USC policy on academic integrity and penalties for cheating and plagiarism. These rules will be strictly enforced.

Problem 1: Geometric Modification (50%)

(a) Geometric Transformation (20%)

Three sub-images "lighthouse1.raw" ~ "lighthouse3.raw" as shown in Figure 1(a) were cut out of an image of a scene containing a lighthouse. The three holes in the image "lighthouse.raw" as shown in Figure 1(b) are all of size 256x256. Each sub-image is from one of the hole in the image, but with different scales and orientations. You are asked to fill the holes in "lighthouse.raw". Write a program to implement a geometric transformation algorithm to create the filled image of size 512x512.



(a) lighthouse1~3.raw (b) lighthouse.raw

Figure 1. Boat

One possible way to do it is:

1) Find the coordinates of corners in each sub-image denoted by $F_i[x,y]$, $i=1,2,3$. This must be done by your program. You will lose points if you do this manually.
2) Design a generic geometric transform $(q_x, q_y) = \Xi_i (p_x, p_y)$ which maps point (p_x, p_y) to point (q_x, q_y) in the i-th sub-image. Here, you need to combine one or more translation, rotation and scaling operations together. After the generic geometric transformation, the transformed i^{th} sub-image should be a square image with its sides aligned with the horizontal and vertical axes.

3) For scaling-up, implement the interpolation function $\Theta(\cdot)$ such that $\Theta(F_i[\Xi_i(x, y)])$ is of size 160 x 160. Drop redundant pixels when scaling-down.
4) Find the coordinates of the holes (specifically the top-left corners). This must be done by your program.
5) Fill $\{\Theta(F_i[\Xi_i(x, y)]), i = 1,2,3\}$ into the holes to get the final image. You can assume you know "lighthouse1.raw" goes to the left hole, "lighthouse2.raw" goes to top, and "lighthouse3.raw" goes to bottom-right.

Hint: Bi-linear interpolation may be needed to generate the pixel value at fractional positions.

(b) Spatial Warping (20%)

You can use the spatial warping technique to create fantastic images. Figure 2 shows an example: We can warp a square-shaped image (on the left) into another shape (on the right).

(a) square-shaped image (b) warped image

Figure 2. Image warping

Develop a spatial warping algorithm as the one demonstrated in Figure 2 and apply it to the 512 x 512 gray scale image "hat.raw" shown in Figure 3 to obtain a warped image. The height of each arc in Figure 2(b) is 128, and the tip of the arc should be located at the center line of the image. Please indicate the reference points to be used and derive the reverse address mapping function that converts Figure 2(a) to (b).

Figure 3. hat.raw

(c) Lens Distortion Correction (10%)

Radial distortion often happens when an image is captured on a non-flat camera's focal plane. The relationship between the actual image and its distortion in the camera coordinate system is as follows:

$$x_d = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$
$$y_d = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \quad [1]$$

where x, y are undistorted pixel locations, K_1, K_2, K_3 are called radial distortion coefficients of the lens, and $r^2 = x^2 + y^2$. The x, y, x_d, y_d are defined in camera coordinate system, to get these values from a given digitized image, you need to convert the pixel locations from the image coordinate (u, v) to the camera coordinate (x, y) as follows:

$$x = \frac{u - u_c}{f_x}$$

$$y = \frac{v - v_c}{f_y}, \quad [2]$$

where (u_c, v_c) is the center of the image, f_x and f_y are the scaling factors. To recover the undistorted image, the (x, y), given the (x_d, y_d), the key lies on finding the inverse function so that:

$$x = f(x_d, y_d)$$
$$y = g(x_d, y_d) \quad [3]$$

However, there is no exact inverse function for Eq. [1] since the forward mapping from (x, y) to (x_d, y_d) is not linear. A common solution is first to project (x, y) to (x_d, y_d) which ends up in triplets of (x, y, x_d) and (x, y, y_d). We then use linear regression (refer to https://www.mathworks.com/help/matlab/data_analysis/linear-regression.html for further readings about linear regression) to find the best approximate inverse functions.

You are given the following distorted image (Figure 4.), and given that K_1 = -0.3536, K_2 = 0.1730, K_3 = 0, f_x = f_y = 600, please implement the aforementioned method to correct the distortion. You can use third party tools (MATLAB, Python Scipy, etc.) for the linear regression part.

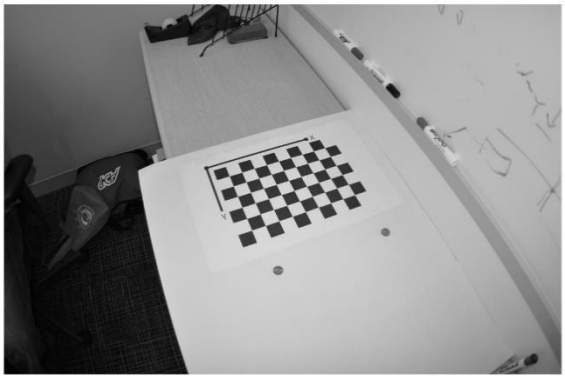


Figure 4. classroom.raw

Appendix:

Problem 1: Geometric image modification

lighthouse.raw	512x512	8-bit	gray
lighthouse1.raw	256x256	8-bit	gray
lighthouse2.raw	256x256	8-bit	gray
lighthouse3.raw	256x256	8-bit	gray
hat.raw	512x512	8-bit	gray
classroom.raw	1072x712	8-bit	gray

Problem 2: Digital Half-toning

pattern1-4.raw	375x375	8-bit	gray
deer.raw	691x550	8-bit	gray
rice.raw	500x690x3	24-bit	color(RGB)

Reference Images

All images in this homework are from the USC-SIPI [1] image database and MPEG-7 Shape Dataset [2].

Reference

[1] http://sipi.usc.edu/database/
[2] http://www.dabi.temple.edu/~shape/MPEG7/dataset.html