BLG 453E Homework - 1

Due 09.10.2017 22:00

Policy: Please do your homework on your own (Do not copy paste your solutions from the internet or your friends). The code and the report you submitted must be your own work. All code must be implemented using **Python** programming language and **OpenCV Python wrapper**.

For your questions: albay@itu.edu.tr

1. **Histogram Equalization**

   NOTE: DO NOT USE built-in library functions for this question, you have to write your own histogram equalization code, otherwise, you will not learn anything, hence get no points from this problem.

   Start with loading the image ”underexposed.jpg”, which is an underexposed true color image. Display the image with the ”underexposed image” title.

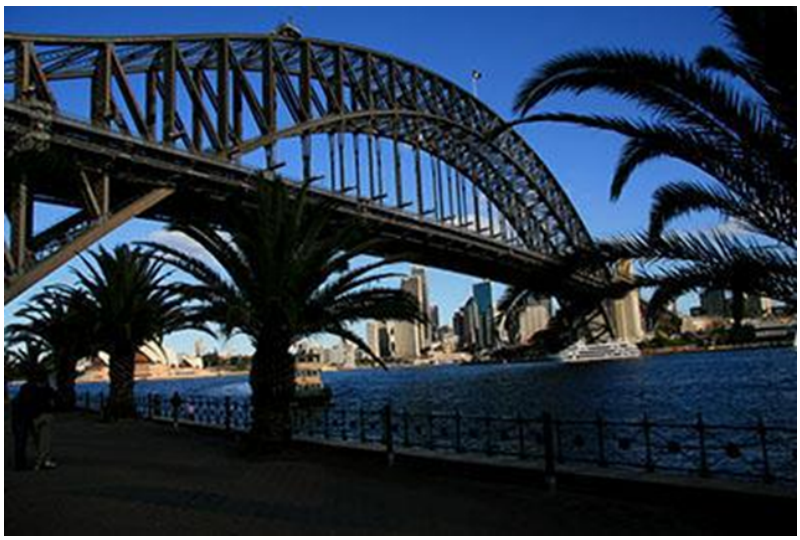

Figure 1

   (a) Calculate the mean and standard deviation over each color channel of the image

   (b) For each color channel of your image compute the histogram and its sample probability distribution function (pdf), i.e. the normalized histogram. Plot your histogram result with the title ”histograms of the underexposed image”. (Hint: here you can use the 1D hist function)

(c) Compute the corresponding cumulative distribution function (cdf) of your image using the pdf. Plot your result with the title ”cdfs of the underexposed image”. (Hint: here use the numpy.cumsum)

(d) Write your own function to equalize the image brightness without creating serious changes in the colors. Display the image with the ”Brightness Equalized Image” title.Compute the histograms of your new image. Plot your results with the title ”histograms of the equalized image”.

(e) Comment on the effect of the above operation on the pdf and on your image, calculate its mean and standard deviations as well.

(f) Give an intuitive explanation to how the generated gray level histogram equalization affects the occurrence rate of gray levels in the output.

2. **Histogram Matching**

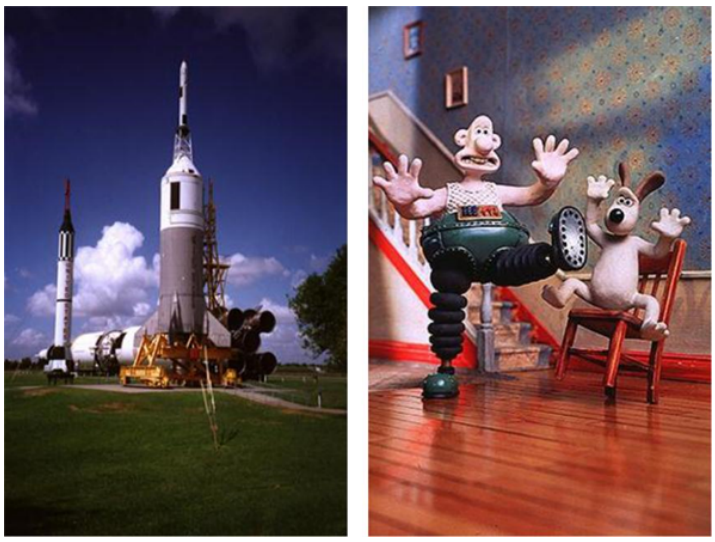

Figure 2

(a) For each color channel of your images (Color1, Color2) compute the histogram and the sample probability distribution function (pdf), i.e. the normalized histogram.

(b) Write your own function to match the histogram of the target image (Color2) to the image that will be remapped (Color1). Display the image with the ”histogram matched image” title. The resulting image should look like below:

Figure 3

(c) Compute the histograms of your histogram matched image. Plot your results with the title ”histograms of the matched image”.

3. **Geometric Transformations: Affine Transformation**

Imagine that you have been given images of the same scene, taken with different cameras. Suppose that the second camera delivers a geometrically distorted image (cameramanAffine.jpg), and that the resulting image can't be directly overlaid on the image generated with the first camera ('cameraman.jpg'). In order to perform any sort of analysis of the images, you must first ensure that the images line up; this can be accomplished using geometric image transformations.
In this assignment, you are to write a Python script that will transform image 'cameraman.png' line up with the image 'cameramanAffine.png'. Your goal is to try to find the optimal geometric transformation between these two images. You will write your own transformation script without using any built in function in Python libraries (Write your own bilinear interpolation function). First determine appropriate point mappings (corresponding points) between each pair of images. For example, you can select eye of the cameraman as a one of the points.

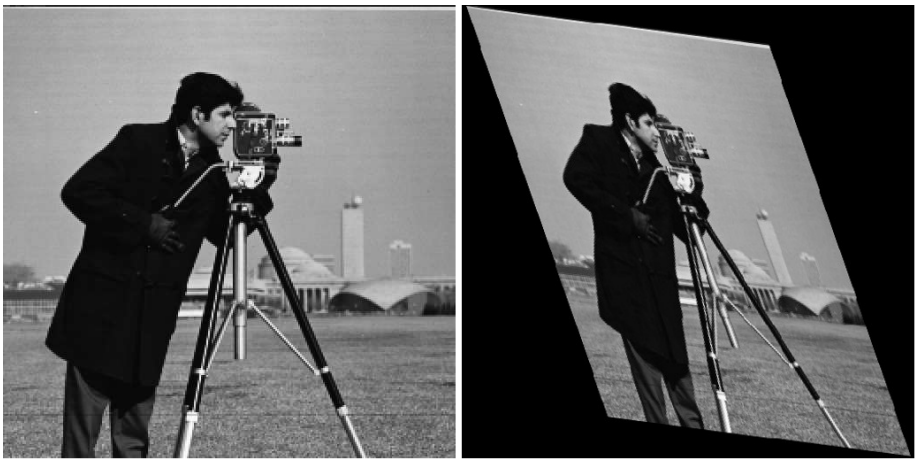

Figure 4

(a) Estimate the 2D affine transformation matrix parameters (4 of them) by using for instance 5 corresponding features from both of the images. You can use bult-in functions to capture mouse click operations for selection of pixels.

(b) Using these 5 point correspondences, set up a simple Linear Least Squares problem $(Ax = b)$ to solve for $a_{11}, a_{12}, a_{21}, a_{22}$ (Affine matrix A's elements).

(c) Compare your result with the cameramanAffine image to check if they look similar, i.e. you are checking visually whether the affine transform you estimated is correct.

(d) Use the estimated A matrix to transform the original image. How well does it line up with cameramanAffine? Try to analyze your transformation results and optimize the selection of the corresponding points.

4. You are given following image. Rotate the image by 30 degrees clockwise with respesct to: (i) image center used as the center of rotation (ii) top left corner used as the center of rotation.

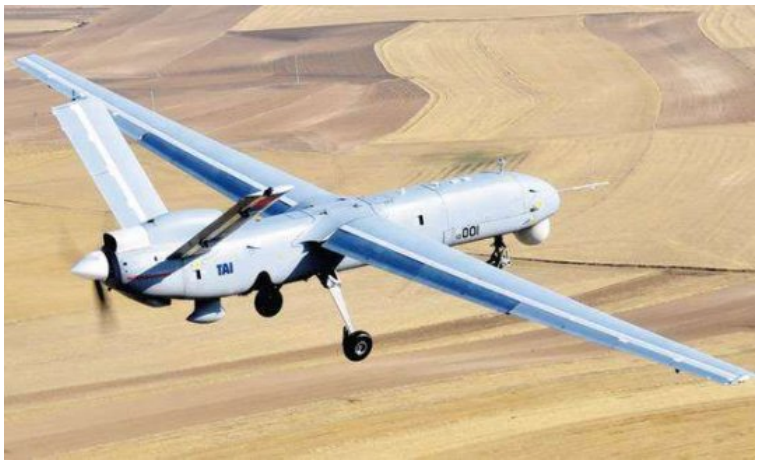

Figure 5

5. You are given following image. Define a nonlinear mapping to produce the swirl effect that is shown below (Hint: To do this convert the image cartesian coordinates to polar coordinates using built-in functions. Sum angle and weighted radius linearly. Use this sum for mapping of coordinates. Use small weights to scale radius.):

Figure 6: (Taken from your lecture slides.)