Middle East Technical University Department of Computer Engineering

CENG 466

Fundamentals of Image Processing

Fall '2018-2019

Take Home Exam 1

Due date: November 1 2018, Tuesday , 17:00

1 Objectives

The purpose of this assignment is to familiarize you with the fundamental spatial domain image enhancement techniques. For each question you are required to develop your own algorithm based on the techniques you learned in the lectures.

2 Specifications

You are given three questions, which you should solve with your own algorithms. In addition to the solutions, you are required to prepare a report that explains your methodology and includes the analysis of the results and your comments on them. The report should be **3-5 pages** long and should be prepared in IEEE Conferance Proceedings Template (LaTeXis recommended) provided in the following link.

`https://www.ieee.org/conferences_events/conferences/publishing/templates.html`

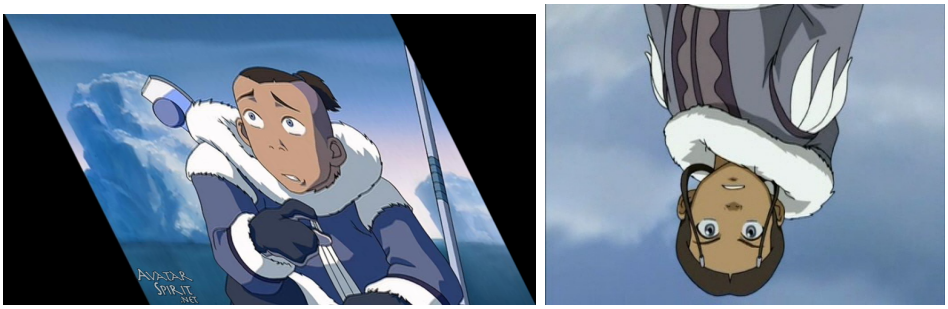
- Grading will be based on the quality of the outputs, script contents and the report
- The report should clearly explain the methodology and rationale behind the algorithm design. It should also explain the difficulties encountered in the design, implementation and experimentation stages, and your solutions on them. Last but not least, the report should contain your comments on the results. Even if the results does not match your expectations you should discuss the encountered situation.
- In your solutions you are **NOT** allowed to use any function from Image Processing Toolbox of MATLAB other than imread or imwrite.

2.1 Question 1 (20 Points) - Geometric Transformations

In this part you are required to identify the degenerations in the given images **A1.png, A2.png, A3.png, A4.png** and **A5.png** shown in Figure 1. These images are modified using some geometric transformations. Your job is to detect them and recover the provided original versions as much as possible. Note that full recovery is not possible in some cases due to the information loss.

Implement your solution as a MATLAB script named **the1_partA.m** which processes the images named AX.png. After running the script, images named **AX_output.png** should be created as the reconstruction.

(a) A1 (b) A2 (c) A3



(d) A4 (e) A5

Figure 1: Modified images of part 1

2.2 Question 2 (40 Points) - Histogram Processing

In this part you are given two images B1.png and B2.png shown in Figure 2. Your job is to enhance their quality by using histogram processing techniques. For this purpose you will implement the following in a MATLAB script named **the1_partB.m**,

- Apply histogram equalization on both images B1.png and B2.png. Your script should create two different images for each input. Create **BX_histeq.png** which shows the equalized histogram, and **BX_histeq_output.png** which shows the input image after histogram equalization.
- Now, assume you have parts of the original image that you can use to manipulate given images. Apply histogram matching on B1.png and B2.png where the referance images are given as B1_ref.png and B2_ref.png respectively. Your script should create two different images for each input. Create **BX_histmatch.png** which shows the matched histogram of the input image BX.png, and **BX_histmatch_output.png** which shows the input image after histogram equalization.
- Discuss your findings, addressing to the similarities or differences of histogram equalization and histogram matching.

2.3 Question 3 (40 Points) - Edge Detection

In this part you are given three images C1.png, C2.png and C3.png shown in Figure 3. Your job is to find edge maps of these images by following these steps.

- Write a convolution function that takes an image and convolves it with a given filter. For this purpose you will write a MATLAB script named **the1_convolution.m**

(a) B1

(b) B2

(c) B1_ref

(d) B2_ref

Figure 2: Modified images of part 2

- Use the following four edge filters (S_x, S_y, R_x andR_y) and convolve the given images with these filters in a MATLAB script named as **the1_part3_edgefilters.m**. Your script should create 12 files (considering you will convolve 3 images with four filters) Name each one as **BX_Sx.png**, **BX_Sy.png**, **BX_Rx.png**, **BX_Ry.png** accordingly.

$$S_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}, S_y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}, R_x = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, R_y = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

- Combine the results that you have obtained in the previous step according to the following formula;

$$BX\_S\_edges_{i,j} = \sqrt{conv(BX, Sx)^2_{i,j} + conv(BX, Sy)^2_{i,j}} \quad (1)$$

$$BX\_R\_edges_{i,j} = \sqrt{conv(BX, Rx)^2_{i,j} + conv(BX, Ry)^2_{i,j}} \quad (2)$$

  where $conv(BX, ??)_{i,j}$ is the gradient at index (i,j) that you have obtained in the previous step. For this part you will write a a MATLAB script named as **the1_part3_edges.m**. Your script should create 6 files containing the total edge maps of the images. Name each one as **BX_S_edges.png** (obtained from Equation 1) and **BX_R_edges.png** (obtained from Equation 2). Comment on the results based on the differences of the previous step and the necessity of this step. **Warning:** You may need to convert the data type to double in order to take the compute the given formula. However, your final results should have a type uint8. Please be careful about type conversions.

- Define a Gaussian filter and convolve given images with this filter in order to blur the given images. For this part you will write a a MATLAB script named as **the1_part3_blur.m**. Your script should create 3 files containing the blurred images. Name each one as **BX_blurred.png**.

- Repeat the second and third steps on the blurred images. In other words extract the edge maps of the blurred images in a MATLAB script named as **the1_part3_blurrededges.m**. Your script should create 6 files (considering you will convolve 3 images with two filters) Name each one as **BX_S_blurred.png** and, **BX_R_blurred.png** accordingly.

- Discuss the differences among edge filters according to your findings.

- Discuss the effect of blurring on finding the edge maps according to your findings.

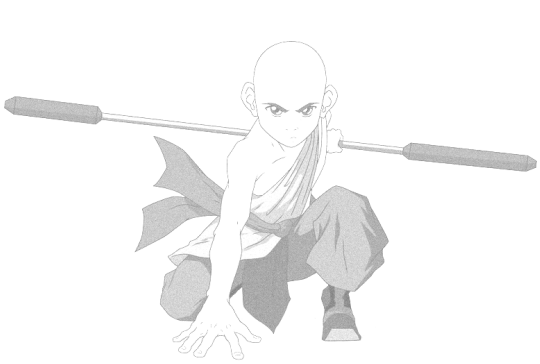
3 Regulations

1. **Group:** You are required to do your assignment in a group of two. If there is an unclear part in your code, we may ask any of the group member to describe that code segment. Also group members may get **different** grades. We reserve the right to evaluate some or all of the groups to determine the contribution of each group member to the assignment.

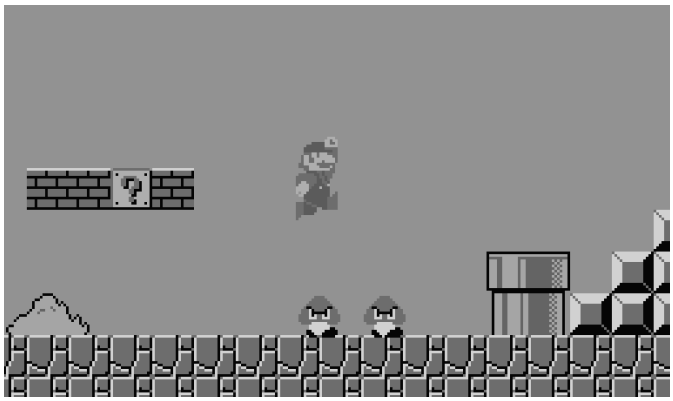
2. **Programming Language:** You must code your program in MATLAB. Your submission will be tested with MATLAB R2018a on department lab machines. You are expected make sure your code runs successfully with MATLAB R2018a.

3. **Late Submission:** Late Submission is **not** allowed!

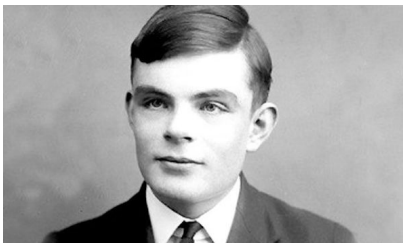
4. **Newsgroup:** You must follow the newsgroup (news.ceng.metu.edu.tr) for discussions and possible updates on a daily basis.



(a) C1



(b) C2



(c) C3

Figure 3: Modified images of part 3

4 Submission

Submission will be done via COW. Create a tar.gz file named THEX.tar.gz that contains all your source code files and the report as a PDF file. Do not send the input and output images. Only one member should submit the homework. Hence, do not forget to **write your names and student id's at the beginning of the scripts.**

5 Cheating

We have zero tolerance policy for cheating. People involved in cheating will be punished according to the university regulations.

Cheating Policy: Students/Groups may discuss the concepts among themselves or with the instructor or the asistants. However, when it comes to doing the actual work, it must be done by the student/group alone. As soon as you start to write your solution or type it, you should work alone. In other words, if you are copying text directly from someone else - whether copying files or typing from someone else's nots or typing while they dictate - then you are cheating (committing plagiarism, to be more exact). This is true regardless of whether the source is a classmate, a former student, a website, a program listing found in the thrash, or whatever. Furthermore, plagiarism even on a small part of the program is cheating. Also, starting out with code that you did not write, and modifying it to look like your own is cheating. Aiding someone else's cheating also constitutes cheating. Leaving your program in plain sight or leaving your computer without logging out, thereby leaving your programs open to copying, may constitute cheating depending upon the circumstances. Consequently, you should always take care to prevent others from copying your programs, as it certainly leaves you open to accusations of cheating. We have automated tools to determine cheating. Both parties involved in cheating will be subject to disciplinary action. [Adapted from http://www.seas.upenn.edu/cis330/main.html]