The Hong Kong University of Science and Technology
Department of Computer Science and Engineering
CSIT 5410 (Spring 2020)

Assignment 1

Total = 100 marks
Due: 11:55pm, March 25, 2020
Assignments must be submitted via CANVAS
Late Policy: 10% reduction; only one day late is allowed, i.e. 11:55pm, March 26

In this assignment, you will use Octave/MATLAB to program two functions *myprewittedge* for image edge detection and *mysiftalignment* for image alignment. You need to complete the missing implementations in the corresponding M-files. From the course CANVAS website, you can download the M-files and related files.

Main Routine: *csit5410_assign1*
The function `csit5410_assign1(FILENAME)` takes a gray scale image with filename `FILENAME`. This routine completes the following tasks:

TASK 1: An image specified by FILENAME is read and displayed.
TASK 2 & 3: The corresponding binary edge image is computed and displayed.
TASK 4: Longest line segment extraction based on Hough transform.
TASK 5: Image alignment using SIFT. You are given four images, "QR-Code.jpg", "image1.jpg", "image2.jpg" and "image3.jpg". Your task is to find the image which matches "QR-Code.jpg" best from the three images.

Function: *myprewittedge*
The function `myprewittedge`, with format `g=myprewittedge(Im,T,direction)`, computes the binary edge image from the input image `Im`. This function takes an intensity image `Im` as its input, and returns a binary image `g` of the same size as `Im` ($m \times n$), with 1's where the function finds edges in `Im` and 0's elsewhere. This function finds edges using the Prewitt approximation to the derivatives with the assumption that input image values outside the bounds are zero and all calculations are done using double-precision floating point. The function returns `g` with the size $m \times n$. The image `g` contains edges at those points where the absolute filter response is above or equal to the threshold `T`.

Descriptions of the function input parameters:
`Im` = An intensity gray scale image
`T` = Threshold for generating the binary output image. If you do not specify T, or if T is empty ([ ]), `myprewittedge(Im,[],direction)` chooses the value automatically according to **Algorithm 1** described below.
`direction` = A string for specifying whether to look for '`horizontal`' edges, '`vertical`' edges, positive 45 degree '`pos45`' edges, negative 45 degree '`neg45`' edges, or '`all`' edges (maximum of all responses).

Algorithm 1: Automatically determined threshold

1. Initialize T to be the mean between the minimum and maximum values in the response map (the map obtained by applying the edge detection algorithm).
2. Threshold the image using T, which produces two sub-regions G1 and G2:
   G1, consisting of all pixels with intensity values >= T; and
   G2, consisting of pixels with values < T.
3. Compute the average intensity values m1 and m2 for regions G1 and G2.
4. Update the threshold value: T=0.5*(m1+m2).
5. Repeat steps 2 through 4 ten times. (Optional: repeat steps 2 through 4 until the change of T in successive iterations is smaller than 5%.)

Function: *mylineextraction*

In this task, function *mylineextraction* is called by the routine csit5410_assign1 for extracting the longest line segment from a binary image. This function takes the edge image `f` from task 3 as its input, and returns beginning and end points of the longest line segment found in `f` based on Hough transform. Your task is to complete the missing implementations in *mylineextraction.m*. You may need the following predefined MATLAB functions to complete the routine of *mylineextraction.m*: `hough, houghpeaks, houghlines.`

Function: *mysiftalignment*

The routine csit5410_assign1 calls function *findBestMatching* in this task, and the implementation of the function is given in *findBestMatching.m*. Function *n=findBestMatching(I,I1,I2,I3)* takes 4 images as input and returns the image index which matches the QR-code image (*I*) best. In this function, *mysiftalignment* is called for finding the number of matching pairs for two images. And your task is to complete the missing implementations in *mysiftalignment.m*. The function *mysiftalignment* aligns two images by using the SIFT features and returns the number of matching pairs after screening out incorrect matches. There are 3 steps in the alignment procedure, detailed descriptions can be found in the *mysiftalignment.m* file. You need to complete the routine of *mysiftalignment.m* based on the given implementation of SIFT feature extraction (*sift.m*), matched keypoints extraction (*match.m*), and screening out incorrect matches (*screenmatches.m*).

For all submitted files in this assignment, you cannot use the following predefined functions: `edge`, `fspecial`, `imfilter, conv, conv2;` otherwise, no marks will be given.

The file sample_output.pdf consists of some figures generated from Octave. You are supposed to obtain similar output on the screen when you run the following commands in the Octave environment:

```
>> csit5410_assign1('fig.tif')
```

Assignment Submission and Marking

1. Your submitted programs must be *myprewittedge.m, mysiftalignment.m, mylineextraction* plus other related M-files (plus README.txt file).
2. You must include a README.txt file indicating the programming software (**Octave or MATLAB**) that you are using for this assignment. By default, we will grade your assignment with **Octave** if the README.txt file is missing.
3. **Runtime requirement**: your program must be able to finish all the above tasks within **10** minutes.
4. You must compress all your files with the following filename format: [your 8-digit student ID]_assign1.rar (or zip), e.g. 09654321_assign1.rar, into one file.

5. If your assignment compressed file has been submitted multiple times before the due date (including late submission date), the new version will replace the old version in marking.

6. Note that we take plagiarism seriously. You are allowed to discuss or share your idea to your classmate, but you are not allowed to share your code/pseudocode of your assignment. Please also follow the referencing skills at https://libguides.ust.hk/referencing/plagiarism to avoid plagiarism.

7. Marks would be deducted if there are any violations of the above requirements.

~~ End of Assignment 1 ~~