1. Software Requirement:

MATLAB

2. Prerequisites:

You may refer to your lecture notes and the textbook.

3. Assignment Requirements:

Give extra attention to report document of this assignment, that is, providing only some piece of MATLAB code does not satisfy requirements. For all questions, you need to provide clear and scientific justifications; otherwise you cannot receive full points from that question.

4. Code Style and Naming Conventions

It is also required that you need to follow a standard coding style.

- ➤ All of the questions must be implemented as separate functions and function names must obey following format; "question_id_letter". For example, if you are writing a code for question-4.a, your function name must be "question_4_a".
- At the beginning of the each function, there must be a comment header describing the general inputs/outputs for your function. (Comment header's format is specified in the file "comment style.m")
- ➤ All functions must be called by a single MATLAB script, whose name is "main.m"
- ➤ All of the output images must be written into a folder named as "outputs". (see: MATLAB's imwrite function)

5. Assignment Questions

1) Linear Spatial Filters – Correlation (10 pts):

In this question, you are going to explore the one of the most important practical implication of correlation, namely template matching.

a) Implement a basic face detection algorithm using template matching method for given input image "faces.png" and template image "face_template.jpg". The algorithm should output filter (correlation) response and detection result. Save the outputs with names "correlation_response.jpg" and "detected_faces.jpg". (see MATLAB's imfilter function for correlation routine) (Note: You are not required to develop a complex face detection algorithm, that is, using only MATLAB's correlation procedure is sufficient. For simple outputs, see Figure-1. Also, realize that detection result is thresholded version of filter response.)

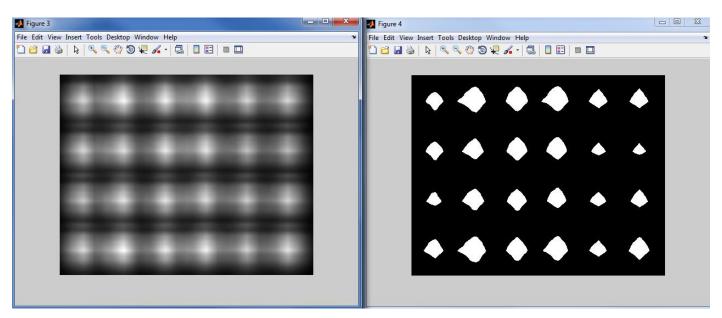


Figure 1: Filter response (lhs) and detection result (rhs).

b) Comment on the results. For example, you should explain the relation between input images (reference and template images) and filter response. Also, you need to explain the peak responses occurred in the output image. Where is the peak response? Why?

- 2) Linear Spatial Filters Sharpening Filters Basics (35 pts):
- a) Crop a thin line (strip) from the given gray scale image ("Assignment_3.jpg"), representing the pixels on the 100th row and columns between 210 and 260. Consider this strip as a signal, and plot the signal's original state with its first and second derivatives (see Figure-2). Save this plotting with name "derivative plotting.jpg".

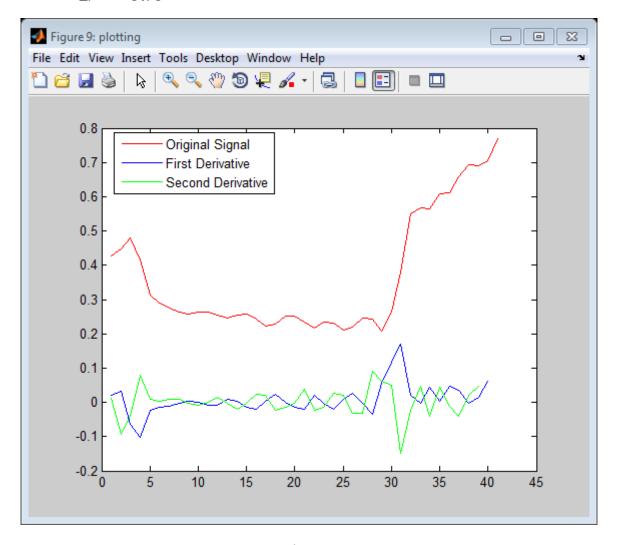


Figure 2

- b) Compare first and second derivatives of the signal from the perspective of sharpening. How can they help us to find intensity transitions (ie. edges)?
- c) You are given two different Laplacian filters, h₁ and h₂. Firstly, filter the image ("synthetic.jpg") with h₁, and then filter it with h₂. Save the outputs with names "synthetic_filtered_laplacian_h1.jpg" and "synthetic_filtered_laplacian_h2.jpg". Observe and comment on the difference. Explain the difference in accordance with derivatives, theoretically.

IS-566 IMAGE PROCESSING ALGORITHMS: Assignment 3 (Due 9 April 2017)

d) Image gradient is the implication of directional change in the intensity. The important thing about the gradient is that it provides direction of the intensity transitions. On the other hand, you can calculate the magnitude of the intensity transition by using several edge detector filters.

At this question, you are given two different Sobel filters, h_1 and h_2 . Firstly, filter the image ("Assignment_3.jpg") with h_1 , and then filter it with h_2 . Save the outputs with names "filtered_sobel_h1.jpg" and "filtered_sobel_h2.jpg". Observe and comment on the difference. Explain the difference in accordance with derivatives, theoretically. How can you combine the outputs to get more promising results?

e) By using the output of filters h₁ and h₂, calculate the gradient's direction in degrees. Save the outputs with name "gradient_direction.jpg". Is it possible to calculate the gradient's direction by using Laplacian filters? Why?

3) Image Noise & Noise Removal (25 pts):

- a) By using MATLAB's imnoise function, add noise to "Assignment_3.jpg". You should create at least three noisy images, and use Gaussian with salt-pepper noise. Save the outputs with names "noisy_1.jpg", "noisy_2.jpg" and "noisy _3.jpg". The parameters of the noise algorithms are up to you.
- b) Devise an average filter by yourselves (you are not allowed to use MATLAB's fspecial function for this question). Then, modify this filter to obtain a weighted averaging filter (you can distribute the weights as you like). Finally, use these filters to clean out your noisy images. Save the outputs with names "smooth_avg_(%img_id).jpg" and "smooth_wavg_(%img_id).jpg".
- c) For this question, you are required to implement two distinct non-linear filters; geometric and harmonic mean filters. Then, use these filters to clean out your noisy images. Save the outputs with names "smooth_geomean_(%img_id).jpg" and "smooth_harmonic_(%img_id).jpg".

 (Hint: You need to use MATLAB's nlfilter for nonlinear filtering)
- d) Mathematically, show that geometric mean filter is nonlinear.

IS-566 IMAGE PROCESSING ALGORITHMS: Assignment 3 (Due 9 April 2017)

4) Composite Spatial Filters - Steerable Filtering (30 pts):

In this section of the assignment you are going to explore the mechanisms of composite filters. In fact, you can craft a filter by using the definitions of continuous functions for a given special task. For this section, your task is to find roads from a given gray-scale satellite image, "roads.jpg".

(**Note**: Several examples of custom spatial filter implementations are provided in a MATLAB script (see "craft_filter.m"). In this script, you can find 5 different filters including cosines, tangent, Gaussian, disk and ideal filters.)

- a) Give mathematical function definitions of the given filters. Also, you need to explain the functionalities of the variables/constants used in these functions.
- b) Craft a composite filter by combining the given (primitive) filters to detect roads. If you want, you can introduce new primitive filters to use in this combination. Your filter must detect the vertical (y-axis direction) roads. Roads are elongated structures that can vary on their widths. Therefore, your filter should provide variability in x-axis in order to extract roads having different width characteristics.

(**Note**: Given filters are already capable of finding roads. However, you need to devise your own filter for this task.)

(**Hint**: You can combine two filters by an element-wise matrix multiplication operator. In fact, any kind of combination is possible; you can combine by summation, weighted summation, subtraction, multiplication or division of the primitive filters.)

Plot your composite filter, save this plotting with name "composite_filter.jpg" and provide comments/discussions on your design, filter's strengths, capabilities, etc.

c) The filter that you have devised previously is only capable of finding the roads on y-axis direction, so it is a type of oriented filter. Fortunately, you can still find the roads having different orientation features by using steerable filtering technique. In this method, the filter is rotated with gradually increasing angle values and applied to the input image outputting intermediate results at each iteration. Then, intermediate outputs can be linearly combined to get a final result. At this stage, the wisest method to combine them is to take maximum values of the intermediate outputs for each pixel location. However, you are encouraged to propose more efficient combination technique, if it is applicable.

Implement a road detection algorithm by using your composite filter and steerable filtering method. Run your algorithm on the given input image "roads.jpg", save your result, which is a binary road mask, with name "road mask.jpg"

(Note: For more information about steerable filtering, please refer to [1].)

(**Hint-1**: Before rotating the filter, padding your filter will provide more promising outputs, see MATLAB's padarray function.)

(Hint-2: You may need to use your filter with different width parameters to detect both thin and

IS-566 IMAGE PROCESSING ALGORITHMS: Assignment 3 (Due 9 April 2017)

wide roads.)

(Hint-3: Use MATLAB's bwareaopen function to remove noises from a binary mask.)

d) Is your algorithm capable of extracting all of the road regions? Which kinds of patterns your algorithm can find? Explain and conclude your work by providing remarks.

6. Submission and Grading Policy

Assignments will be submitted via METU-Online. Create a rar or zip archive containing plain source codes (*.m files) and the document explaining your algorithms, implementations, results and discussions. Please pay attention on the documentation of your assignment.

The assignments are due at 23:59 PM on the deadline.

Any late submission will have 10% deduction for each late day.

No submissions will be accepted 2 days after the deadline.

You are expected to work individually NOT in groups. You will also be expected to follow the academic integrity rules.

Policy for Copying: Passing the work of others (either from another student or a code on internet etc.) off as your own is a breach of academic ethics and also of the University's disciplinary rules. Students are expected to work on the homeworks individually, not as a group. When you submit a work, it automatically implies that you claim the ownership of the work.

Note that METU is subscribed to some tools which allow cross checking of submitted works as well as checking with any work on internet or any university subscribed to the system. No exceptions will be allowed and any work found to be copied will result in failing the course.

Please send your questions about the assignment to **ucinar@metu.edu.tr** or you may post a message to the forum on ODTU-Class.

REFERENCES

[1] Freeman, W.T.; Adelson, E.H.; , "The design and use of steerable filters," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.13, no.9, pp.891-906, Sep 1991

doi: 10.1109/34.93808

(URL: http://people.csail.mit.edu/billf/papers/steerpaper91FreemanAdelson.pdf)