

Homework 3: Fourier, Sobel, Transformations, and Registration

INSTRUCTOR: DANIEL L. PIMENTEL-ALARCÓN

DUE 03/24/2020

GO GREEN. AVOID PRINTING, OR PRINT 2-SIDED MULTIPAGE ;)

3.1 Fourier & Sobel

In the first section of this homework you will learn about the Sobel filter, and investigate some Fourier transform properties. In this section you may use Matlab built-in functions.

- (a) The Sobel filter approximates the gradient of an image using a pair of 3×3 kernel filters:

$$\mathbf{K}_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad \mathbf{K}_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}.$$

\mathbf{K}_x estimates the gradient in the horizontal direction, and \mathbf{K}_y estimates the gradient in the vertical direction. The magnitude of the overall gradient is the sum of both. This is a very popular edge detection filter. Load the `liver.jpeg` image from the course website. Use `subplot` to display:

- The original image \mathbf{X} ,
 - The result of convolving \mathbf{X} with \mathbf{K}_x , i.e., $\mathbf{X} * \mathbf{K}_x$,
 - The result of convolving \mathbf{X} with \mathbf{K}_y , i.e., $\mathbf{X} * \mathbf{K}_y$,
 - The sum of both convolutions, i.e., $\mathbf{X} * \mathbf{K}_x + \mathbf{X} * \mathbf{K}_y$.
- (b) Use `subplot` to display the original Image \mathbf{X} and its Fourier transform, with lower frequencies at the center, and properly log-scaled.
- (c) We will now study the effect of convolution in the Fourier domain. Letting $\hat{\mathbf{X}}$, $\hat{\mathbf{K}}_x$ and $\hat{\mathbf{K}}_y$ be the Fourier transforms of \mathbf{X} , \mathbf{K}_x and \mathbf{K}_y (properly padded with zeros), compute and display (using `subplot`):
- $\hat{\mathbf{X}} \odot \hat{\mathbf{K}}_x$,
 - $\hat{\mathbf{X}} \odot \hat{\mathbf{K}}_y$,
 - $\hat{\mathbf{X}} \odot \hat{\mathbf{K}}_x + \hat{\mathbf{X}} \odot \hat{\mathbf{K}}_y$.

Is there some pattern that you can recognize? E.g., do \mathbf{K}_x and \mathbf{K}_y behave like low-pass or high-pass filters?

- (d) Compute and display (using `subplot`) the inverse Fourier transform of the matrices in (c). How do these compare with the direct convolutions from (a)?

3.2 Spatial Transformations & Landmark Registration

In this section you will implement some basic spatial image transformations (translations, rotations, and scalings), and align two images using Landmark registration. In this section you may use any interpolation method you want.

- (e) Create your own function to translate an image.
- (f) Create your own function to rotate an image.
- (g) Create your own function to scale an image.
- (h) Load the `liver.jpeg` image from the course website. Using your functions from above, rotate it 40° , translate it (30,20) pixels, and scale it by 1.5. Use `subplot` to display the original image \mathbf{X} and the transformed image \mathbf{X}_T at each stage, for all the possible transformation orders. Are they the same?
- (i) To perform Landmark registration, select $K = 10$ reference points (x_k, y_k) in \mathbf{X} , and their corresponding points (x'_k, y'_k) in \mathbf{X}_T . *Hint: Determining such points should be easy, as you had control over the spatial transformation.* Use these points to identify the mapping matrix \mathbf{M} that aligns \mathbf{X}_T with \mathbf{X} . What do you obtain for \mathbf{M} ?
- (j) Create a function that applies a mapping \mathbf{M} to an image. Use `subplot` to display \mathbf{X}_T and the result of applying the mapping \mathbf{M} you obtained above to \mathbf{X}_T . Is it the same as \mathbf{X} ?
- (k) What happens if instead of $K = 10$ you use $K = 5, 3, 2, 1$ reference points? What conclusions can you draw from this?