COMP0114 Inverse Problems in Imaging. MiniProject

This assignment is designed to allow you to learn about the Radon Transform and its inverse. There are two parts : Part A should be done by everyone; it carries 60% of the marks. In part B you have a choice of advanced topics. You should pick *one* as an individual miniproject; this carries 40% of the total marks.

Part A : Core Project. **60%**

1. Calculate the Radon transform of an image and test the back-projection method.

- Load an image of the Shepp-Logan phantom of size 128×128. We will refer to this as $\boldsymbol{f}_{\mathsf{true}}$. Show a picture of $\boldsymbol{f}_{\mathsf{true}}$.
- Generate the Radon transform $\boldsymbol{g} = \mathcal{R}\boldsymbol{f}$ of this phantom in 1-degree intervals from $0-179$. Display $\boldsymbol{g}$ as a 2D-image; this is referred to as the *sinogram* of $\boldsymbol{f}_{\mathsf{true}}$. What is the size of this sinogram and how is this determined?
- Compute the *unfiltered* back-projection and apply it to the sinogram data you generated. What is the size of the back-projected image?
- Compute the *filtered* back-projection and apply it to the sinogram data $\boldsymbol{g}$ that you generated. Verify that this gives a good estimate of the inverse of the Radon transform.
- Add noise to the data $\boldsymbol{g}$ and test how the error in your reconstruction grows with the scale of the measurement noise.

2. Calculate an explicit matrix form of the Radon transform and investigate its SVD.

You can construct an explicit matrix form of the Radon transform by calling it on an image with a single pixel of value 1 and the rest 0.

To do this :

- Create a zero matrix A with number of rows equal to the number of angles $\times$ number of projection samples, and with number of columns equal to the number of pixels in the image.
- Go through a loop for the number of pixels in the image (in column-major). For each pixel j in the loop create an image of the same size with a 1 in that pixel and zero everywhere else. Take the Radon transform of this image and reshape the result into a column vector. This forms the j^{th} column of the matrix A.

Carry out the above steps for a manageable size of matrix, e.g. a 64×64 image and only 45 projections. Take the SVD of this matrix explicitly and investigate how the SVD spectrum (the singular values) varies with

i) Keeping the range of angles from $0 \rightarrow 180$ but varying the number of projections. E.g `angles = [0:4:179]` gives 45 projections at 4 degree seperation.

ii) Keeping the number of projection the same, but varying the range to being less than $0 \to 180$ (i.e. *limited angle*). E.g. `angles = [0:1:44]` gives 45 projections over only a 45 degree range.

3. Implement a matrix-free regularised least-squares solver for the Radon Transform.

You can find a regularised solution to the inverse Radon Transform by solving

$$\left(\mathsf{A}^{\mathsf{T}}\mathsf{A} + \alpha\mathsf{L}\right)\boldsymbol{f}_* = \mathsf{A}^{\mathsf{T}}\boldsymbol{g}$$

where L is a regularisation matrix. For a small problem this can be done with the explicit matrix, but for a larger problem you should implement the matrix-free Krylov solver as you did in coursework two. **Note** : The major difference is that the forward and adjoint operators are not the same.

Compare your solution to the filtered-backprojection method for the cases:

i) full range but small number of angles,

ii) limited angles, as you did in Task 2.

Use both zero-order and first-order Tikhonov regularisation. Remember to add noise to the data.

4. Write a Haar wavelet denoiser.

In this task you will perform denoising using a wavelet transform and shrinkage of the wavelet coefficients.

- Take any (monochrome) image of your choice. Calculate the Haar wavelet transform of this image. Plot some of the coefficients and explain what you see.
- Reconstruct the image from the coefficients by calling the inverse wavelet transform. Check if your reconstructed image coincides with the original.
- Write a function that implements thresholding for a given range (the different scales of your wavelet coefficients) and threshold parameter, and form a modified image by performing the inverse wavelet transform on the thresholded coefficients.
- Create a noisy version of your original image and perform denoising by thresholding of the wavelet coefficients. Investigate the effect of changing the range and the threshold parameter.

5. Iterative soft-thresholding for X-ray tomography.

The aim of this exercise is to write your own sparsity promoting reconstruction algorithm for X-ray tomography, by solving

$$\frac{1}{2}\|\mathsf{A}\boldsymbol{f} - \boldsymbol{g}\|_2^2 + \alpha\|\mathsf{W}\boldsymbol{f}\|_1 \;\to\; \min, \tag{1}$$

where W denotes the wavelet transform. To achieve this we use iterative soft-thresholding with Haar wavelets.

You need to implement an iterative algorithm that performs the update equation:

$$\boldsymbol{f}_{k+1} = S_{\alpha,\mathsf{W}}(\boldsymbol{f}_k - \lambda \mathsf{A}^\mathsf{T}(\mathsf{A}\boldsymbol{f}_k - \boldsymbol{g})). \tag{2}$$

Here $S_{\alpha,\mathsf{W}}$ is the soft-thresholding operator given in lectures,

$$S_{\alpha,\mathsf{W}}(\boldsymbol{f}) = \mathsf{W}^{-1} S_\mu \mathsf{W} \boldsymbol{f}.$$

This process is equivalent to the denoiser from Task 4, where S_μ is given by your threshold function with threshold parameter $\mu = \alpha\lambda$ as discussed in the lecture.

In order to have a full algorithm you need consider the following points:

- Define an initial iterate $\boldsymbol{f}_0$
- What is a good stopping criterion for your algorithm?
- You need to choose a stable step size λ: Start with a small value $\lambda \ll 1$ and make sure that the reconstruction gets better each step. Increase it gradually to converge faster.
- Can we include more prior knowledge, like non-negativity?

Evaluate your algorithm for varying noise levels and projection geometries (low number of angles/ limited angle).

Part B : Advanced Topics **40%**

The advanced topics are in a seperate PDF. This will be handed out in week 10.

Report

You should write a short report, probably no more than 8-10 pages, covering your work for both parts A and B. Indicate clearly which extension in part B you have attempted. In the report you should describe the steps you took and the design choices you made. Indicate any problems encountered and how you solved them. Compare and contrast the different solutions you obtained, and draw conclusions regarding what you consider to be an optimal approach. Code, if you consider it relevant, can be attached as an appendix (i.e. not included in the 6-8 page report).