Concordia University

Department of Computer Science & Software Engineering

COMP 478/6771 Image Processing

Assignment 3 - Due Date: *Nov 15, 2024*

Part I: Theoretical questions

1. (**16 points**) Prove the validity of the following properties of the Radon transform:

$$g(\rho, \theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x, y)\delta(x\cos\theta + y\sin\theta - \rho)dx\,dy$$

(a) (**8 points**) *Linearity:* The Radon transform is a linear operator (use definition of linearity).
(b) (**8 points**) *Translation property:* The radon transform of $f(x - x_0, y - y_0)$ is $g(\rho - x_0\cos\theta - y_0\sin\theta, \theta)$.

Part II: Programming questions (28 points)

1. (**10 points**) Download the image "*cameraman.tif*" from the assignment package. We will implement the adaptive filtering for image denoising discussed in class:

$$\hat{f}(x, y) = g(x, y) - \frac{\sigma_N^2}{\sigma_L^2}(g(x, y) - m_l)$$

Here, $g(x, y)$ is the noisy image, $\hat{f}(x, y)$ is the denoised image, $m_l = \frac{1}{MN}\sum_{s,t\in S(x,y)} g(s, t)$ is the local mean of an $M \times N$ image patch, σ_L^2 is the local variance of the same $M \times N$ image patch, and σ_N^2 is the noise variance. Note that usually, we don't know the variance of the true noise, as a result, we will define $\sigma_N^2 = mean\ of\ all\ compuated\ m_l\ across\ the\ image\ g(x, y)$.

For this question, we will stick with M=N=5.

1) A semi-finished MATLAB script (*adaptivefilter.m*) is provided in the assignment folder for your reference to denoise a noisy image with adaptive filtering. Please complete the code, copy-paste your finished script to your assignment report, and showcase the image before and after image denoising.
2) Filter the image with added Gaussian noise with a 5x5 box filter. Compare the result against that from the adaptive filter with comments.

2. (**18 points**) Download the image "*wheel.png*" from the assignment package then perform edge detection using existing MATLAB functions (with the parameter choices of your own, you may also use equivalent functions in scikit-image) for:

   a) Laplacian of Gaussian (Marr-Hildreth) edge detector

   b) Canny edge detector

   1) (**4 points**) Briefly list the steps involved in implementing the edge detectors.
   2) (**4 points**) Explain how edge linking (the final step of the **Canny algorithm**) was implemented. Does the first method need this step?
   3) (**4 points**) List the parameters that determine the performance of the algorithms. What parameter values did you use and why?
   4) (**6 points**) Show and compare the results obtained by the two methods (give some comments).