# EE420 – Digital Image Processing

# Homework 4

Assigned: Feb 18, 2021 Due: Mar 02, 2021

Maximum Possible Grade: 45 points

# General guidelines

Please upload your response to Canvas as a ZIP file with the following filename convention

* Name of ZIP file FIRSTNAME\_LASTNAME\_Homework4.zip
* Replace FIRSTNAME with your first name
* Replace LAST NAME with your last name

**Failure to adhere to the filename convention will result in deduction of points.**

Provide a detailed response to each question, including supporting mathematical arguments. Include screenshots of image before and after image processing. Failure to do will result in deduction of points.

# Submitting your solutions

Please ensure that the ZIP file uploaded to Canvas, includes the following components:

* PDF file of your write-up including response to all questions.
* Completed MATLAB Source code for PMDiffusion.m
* Completed MATLAB Source code for HWK4\_Diffusion.m

**Failure to adhere to the filename convention will result in deduction of points.**

# Objective

The assignment is designed to help you appreciate the significance of edge preserving filtering using Perona & Malik’s Anisotropic Diffusion model [1,2]. Their pioneering work laid the foundation for an assortment of methods with the purported ability of smoothing over noise in an image while preserving edges. Bilateral filtering is one such example.

You will be using the **cameraman** image and the **colored chips** image included with MATLAB to complete this assignment. The starter code included with the assignment already invokes these images.

The best approach to understanding and appreciating the edge-preserving character of anisotropic diffusion (anisotropic filtering) is to examine an edge map derived from the diffused image (filtered image) at different time instances. An example of such a representation is shown below for Isotropic diffusion using the standard heat equation.



The edge maps can be obtained by invoking MATLAB’s built-in function **edge(I,‘canny’)**. Please read the MATLAB documentation before using the function. The topic of Canny edge detection will be covered in a later lecture.

The starter code provided with the assignment implements Isotropic diffusion using the heat equation. Your task is to implement the PM Anisotropic Diffusion model. You will periodically poll the output, and compute the edge maps associated with the filtered images at different time instants. Run the starter code to see what I mean.

Your implementation of PM must accept the following parameters

* the starting temperature distribution or image
* edge threshold
* time step between successive diffusions (NOTE: for stability)
* the end time . Pick so that it is an integer multiple of
* intermediate times at which we would like to poll the output. Pick entries in such that they are integer multiples of .

Please bear in mind that there are two variants of PM’s anisotropic diffusion. Please implement the exponential diffusivity model . The parameter serves as an “edge threshold”, selecting edge strengths that are preserved. is the gradient magnitude observed at a spatial location.

Recommendations:Please read references [1,2,3] before attempting the assignment. Equations (5),(6) and (7) in [3] are the key equations you will need to implement PM diffusion.

Warning:The original PM paper does not provide the complete numerical recipe for implementing PM diffusion. **Do not attempt to duplicate or modify PM implementations from the internet as they all likely implement an approximation to PM and not the complete PM diffusion.**

I strongly recommend replicating the approach described in the *Numerical Implementation* section of [3]. Although, the notation in [3] is different from the one used in the PM paper, it is easier to implement.

* To implement Eq.(5) in [3], first compute the gradient magnitude at each pixel in the image at the time instant using the built-in MATLAB function **imgradient**. Try **gMag = imgradient(ut,’central’)**. The image **gMag** contains the gradient magnitude at each pixel in the image . [3] refers to **gMag** as
* Now evaluate the Perona-Malik diffusivity at the pixel as
* Eq(6) in [3] requires us to compute the diffusivity at pixels . This may be done as follows
* Exercise caution when evaluating the following terms in Eq(5), as the row column indices may fall outside the image boundary.

A simple approach to computing without encountering errors in MATLAB is provided below

**u([2:m,m],:) – u**

where **m,n** represent the number of rows and columns in the image **u**. The process implicitly assumes Neumann boundary conditions. Replicating the last row allows us to accommodate boundary conditions. You can do something similar to accommodate boundary conditions when computing the remaining terms , and .

**Deliverables & Questions**

1. Completed MATLAB code for PMDiffusion.m (15 points)
2. Completed MATLAB code for HWK4\_Diffusion.m (5 points)

* Code is missing the computation of edge maps and display of edge maps.

1. Read references [1,2] and comment on the principal motivation of Perona and Malik’s paper. (HINT: It is not filtering.) (5 points)
2. For a fixed end time , experiment with diﬀerent values of in PM diffusion and observe the impact on filtering, specifically edges.

Include screenshots of images and edge maps for the **cameraman** and **colored chips** images. Comment on your observations. Is there a preferred value of for each image? (10 points)

1. For a fixed end-time compare the output of Isotropic and PM diffusion. Include screenshots of images and edge maps. (5 points)
2. Experiment with 2 other images of your choice. Explain your motivation for choosing these images.

(5 points)

**References**

1. Perona, P. and Malik, J., 1990. ***Scale-space and edge detection using anisotropic diffusion***. IEEE Transactions on pattern analysis and machine intelligence, 12(7), pp.629-639.

Link: <https://pdfs.semanticscholar.org/9bc8/7686980028e863af29f5136a957cf0dde29f.pdf>

1. Perona, P., Shiota, T. and Malik, J., 1994. ***Anisotropic diffusion***. In Geometry-driven diffusion in computer vision (pp. 73-92). Springer, Dordrecht. (included with homework. DO NOT DISTRIBUTE)
2. File **PM\_Diffusion\_Notes.pdf** included with assignment.