Math 354: Mathematical Modeling Final Project Installment I

Dr. Fred Park

Whittier College

Name:____

Group Members:____

Group Name:____

Note: For full credit you must show all work. Incoherent work without logic or reason will not receive any credit whatsoever! Only your notes from class and textbook are to be used as resources.
Exam is Due by 11:59 PM on Friday December 13th by digital submission in a single PDF file and a zipped folder containing all code used. Any late exams will not be graded! Please document the work done by each individual student in the group. This must be explained in the exam.

This is installment I of the final. The second part will be released shortly.

Let $f = f(x, y)$ be an image that is corrupted by noise. Consider the two image denoising models we discussed in class:

- Total Variation Denoising Model:

$$\min_u J[u] = \min_u \frac{1}{2} \int_\Omega (f-u)^2 d\,x + \frac{\lambda}{2} \int_\Omega |\nabla u| dxdy \tag{1}$$

- Perona Malik Energy:

$$\min_u J[u] = \min_u \frac{1}{2} \int_\Omega \log\left(1 + \alpha |\nabla u|^2\right) dxdy. \tag{2}$$

1. Calculate the functional gradient for the TV model. Include boundary conditions as viewed in class using the vector form of Green's Theorem. What is the condition on u on the boundary?
2. Write out the gradient descent scheme for the TV model. i.e. write out the PDE that results from applying gradient descent to the functional to find the optimal minimum. What are the initial conditions for each? i.e. what is $u(t = 0, x, y) =$?
3. Implement the gradient descent scheme for the TV model in matlab using the cameraman image. Follow the steps below:
   - Use the function '`f = imnoise(f,'gaussian',mean,sigma^2)`' to add a random Gaussian noise of variance '`sigma^2 = 0.003`' with '`mean=0`' to the image. Note: variance $= \sigma^2$ where σ is the standard deviation of the Gaussian distribution of the noise.
   - Use this noisy image f as the initial condition for the gradient descent scheme for both energies.
   - Note: if you end up with a term like $1/|\nabla u|$, you must regularize it so it doesn't blow up. Use the following method: $1/|\nabla u + \delta|$ for $\delta = 0.01/3$.

4. Add noise to a grayscale version of one of your images as in 3. and apply TV denoising to remove the noise. Apply it channel by channel to the color version of your image to remove noise. Here, you would use the imnoise command directly on the color image and then denoise each channel separately. Then put the denoised image channels back into a 3-channel color image.

5. If you were to deal with a type of noise called salt and pepper noise where pixels are randomly flipped to either 0 or 255, what would you do?
   Type the following commands into matlab:
   ```
   I = imread('eight.tif');
   J = imnoise(I,'salt & pepper', 0.02);
   figure, imshow(I), figure, imshow(J)
   ```
   What do you see? How would you deal with this type of noise? Would local averaging work? Why or why not? Use your code from the previous methods to denoise the given coins image you generated with salt and pepper noise. How well did they work?

6. In the TV denoising model, replace the term $\frac{1}{2}\int_\Omega (f-u)^2 d\,x d\,y$ by $\int_\Omega |f-u| d\,x d\,y$ and use this to try and solve the salt and pepper denoising problem. Calculate the functional gradient and write out the gradient descent scheme for this new model. It is called the TV-L1 model. Apply this to the coins image corrupted by salt and pepper noise found in 6.