EE420 – Digital Image Processing – Homework 4

Deliverables & Questions

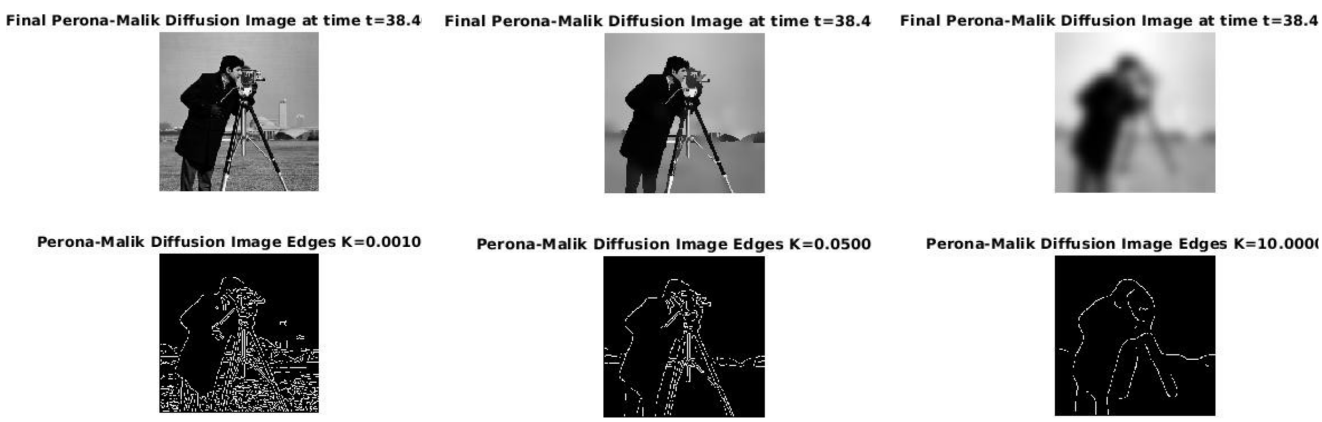
1. The MATLAB code for ‘PMDiffusion.m‘ is attached to this PDF in a ‘/code’ folder.

2. The MATLAB code for ‘HWK4_Diffusion.m‘ is attached to this PDF in a ‘/code’ folder.

3. The principal motivation of Perona and Malik’s papers is the fact that when using a standard scale-space paradigm, the location of a boundary at a coarse scale is not directly available in the coarse scale image, and the locations of the edges at the coarse scales are shifted from their true locations, and the edge junctions, which contain much of the spatial information of the edge drawing, are destroyed. This result in a spatial distortion of the image. In the paper, Perona and Malik are trying to find a method to obtain the true location of the edges, in order to not loose information.

4. For a fixed end time T_{end}, I experimented with different values of K in PM diffusion.

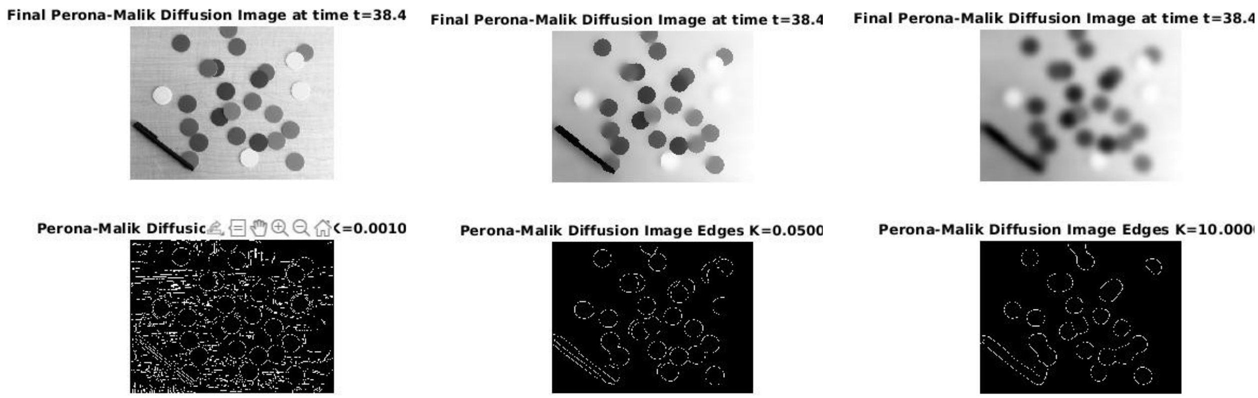
   For the cameramen image:



   We can see that small K makes the algorithm to detect more edges, even for irrelevant details. As the K get larger, we will detect less edges and we will have less details on our diffused image.

   After experimenting several K values, the best K for this image is K=0.05, as can see above. We can clearly see the edges of the cameramen, but we don’t have irrelevant edges and noise in the background.

For the colored chips image:

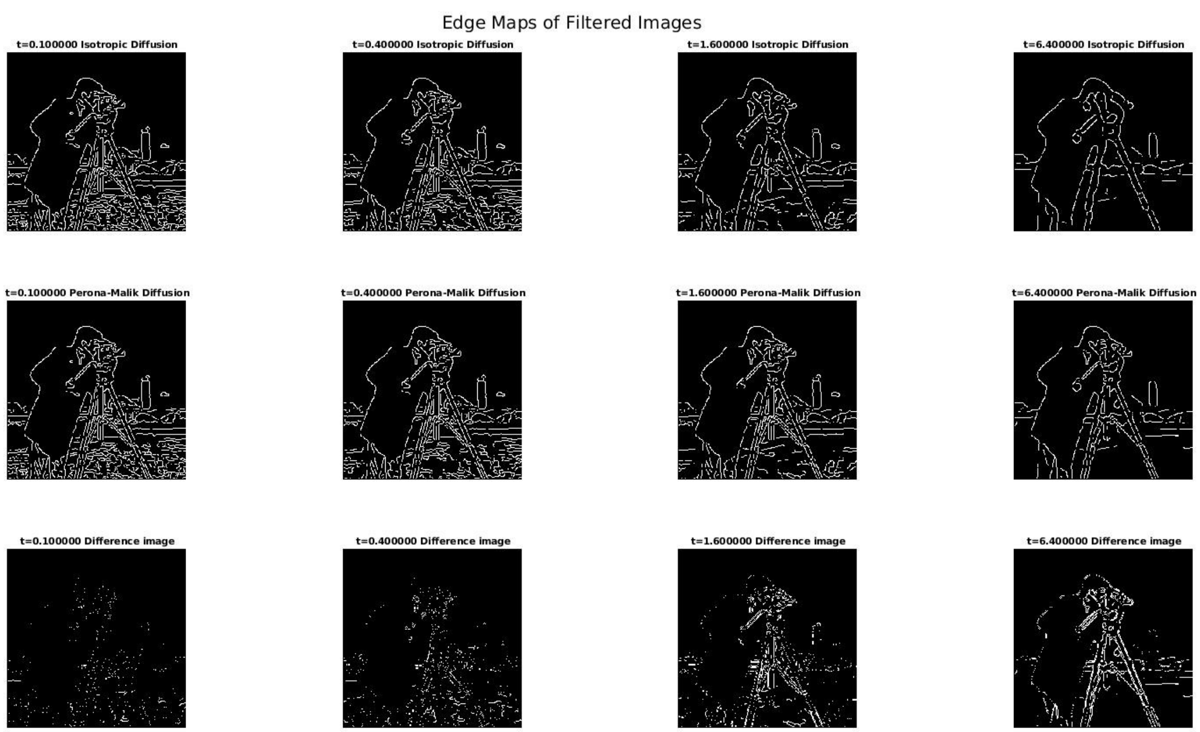


In this image, we can see even more the effect of K size on the image - small K makes the algorithm to detect more edges, up to a point that we cant detect the chips. As the K get larger, we will detect less edges and we will have less details on our diffused image, and the chips connecting to each other.

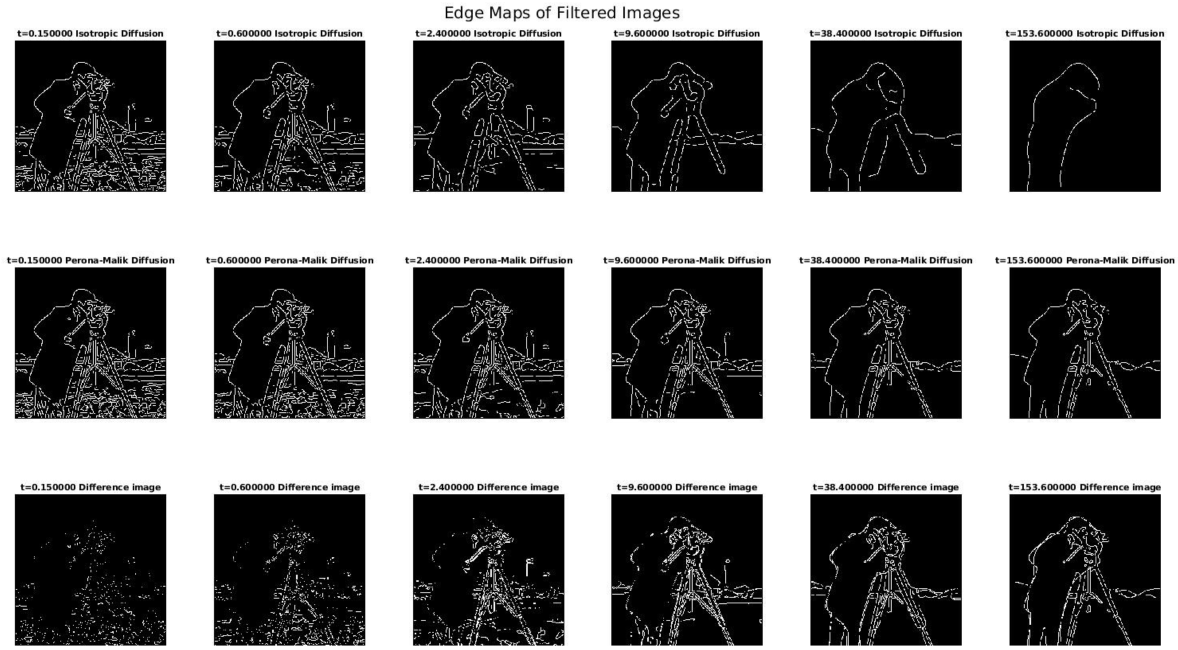
After experimenting several K values, the best K for this image is also K=0.05, as can see above. We can clearly detect the different chips.

5. The following images shows the results for fixed end time T_{end}, using Isotropic and PM diffusion:

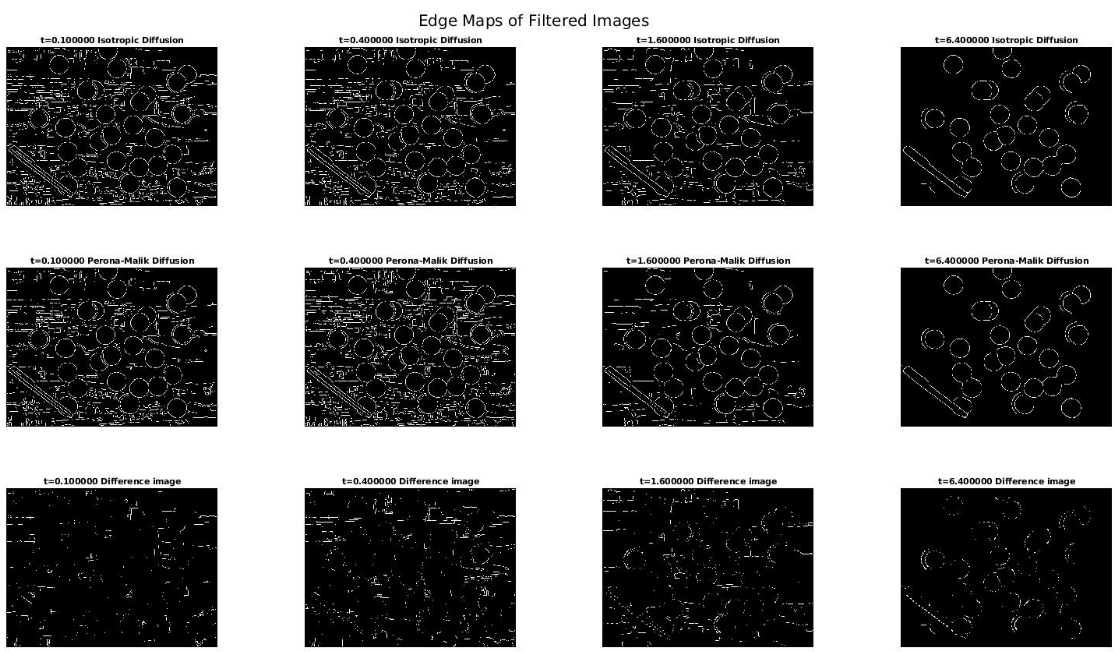
   For the cameramen image:



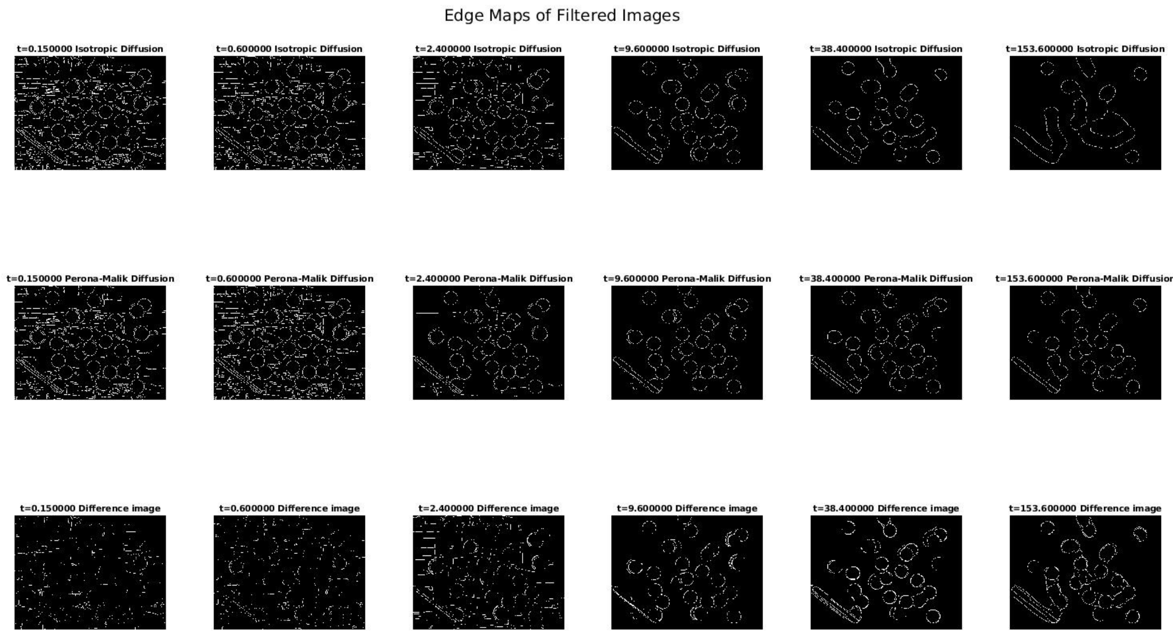
From this image we can see, as the time passes the Isotropic diffusion losses more information in the edge detection process – after ~6.5 seconds, the top image starts to loss information, while the middle image stays detailed (PM diffusion), and the different between the two images grows (we can see that from the difference image in the bottom row). If we will wait more, we will see that we can't detect the edges in the Isotropic diffused image as well as in the PM diffused image, as you can see from the following image in time Tend = 153.6:



For the colored chips image:



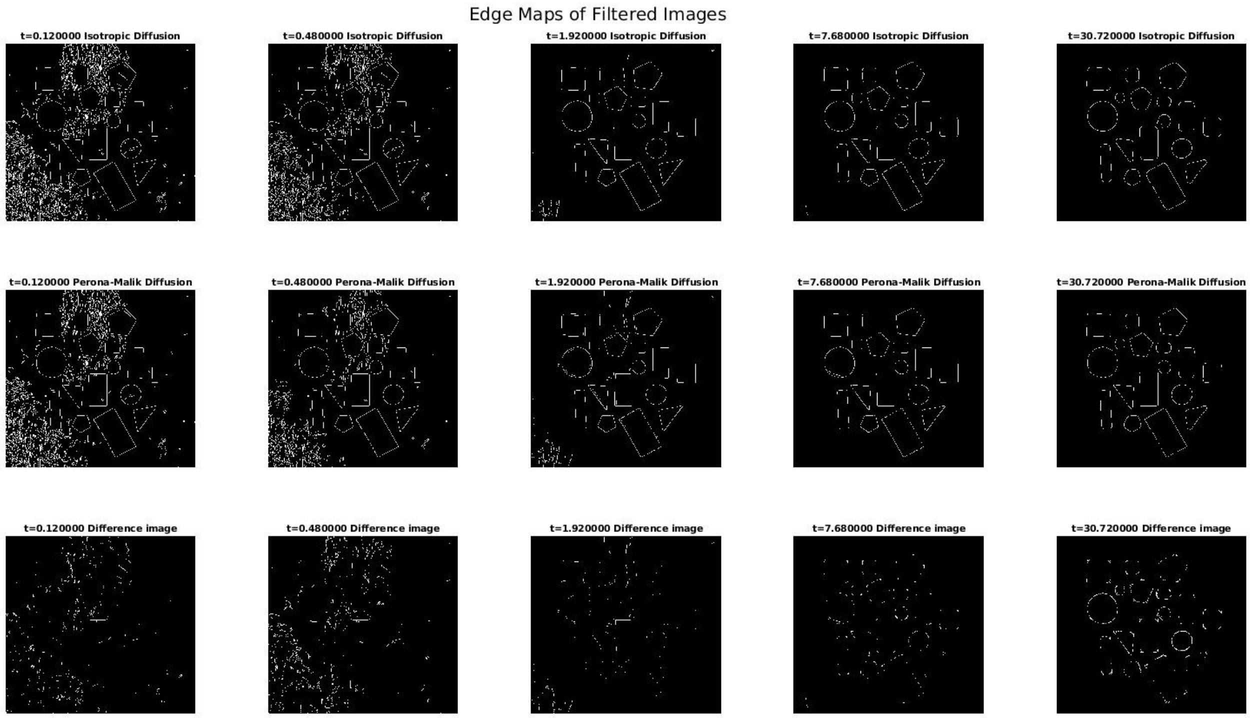
From this image we can barely see the difference between the images, but again, if we will increase the Tend time to 153.6, the difference will be clear:



As we can see, for Tend = 153.6 sec, we can still detect the edges in the PM diffused image, but the Isotropic diffused image is unclear, and we are missing a lot of information (to the point we can't identify what was the original image from the filtered image).

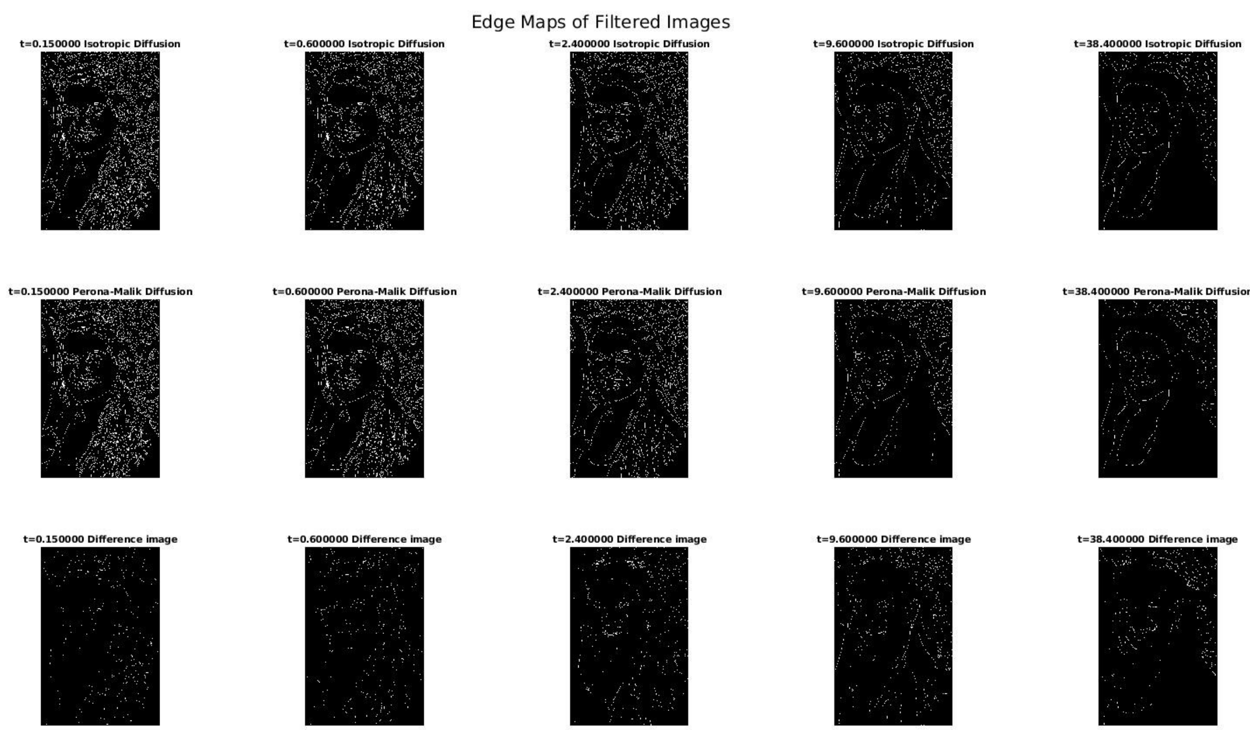
6. In this part, I chose 2 images to apply the diffusion on:
   First image – Edge detection for different shapes:

I chose this image because it had distinct edges in my eyes, but I wanted to see if I can detect those edges easily. The problem it this image is that it's background interfere with the detection – it's hard to find K value that doesn't show the noisy background, and can detect the edges perfectly:



We can see pretty good detection of the edges with K = 0.09 after ~30 second.

Second image – My image:



I chose this image just for fun. As you can see – this image s not ideal for edge detection – it's hard to detect the edges and to separate the person from the background.

In this case, as like all the other cases I presented, the edge detection works better with the PM diffusion (the second image):