Nicolas Graves, Joshua Krchnak, Jinman Cai, Zachary Brewer, Austin Metcalf, Okeith Percy, Harshesh Shah

Image Restoration Report from Group-R

The objective of the group was to take distorted images and to restore them back to their original form. Image restoration normally focuses on both removal of noise from an image as well as blur, but in the time, scope allowed for the project the program created by the group only focuses on noise removal and not blur. The program allows a user to select an image and the type of noise they would like to add to the image. Once a certain type of noise had been selected the user can then select the specific filter that they wanted to use to remove that noise. The types of filters that were available for selection differed based on the type of noise selected. This is because certain filters are only effective against certain types of noise. Once the user has selected both the type of noise and filter, they can run the filter by hitting the "Filter" button in order to see the results. The user can also add values to the parameters of the filters to see how they would affect the filtering of the image. Once the filter has been run the program displays four different images. The first image is the image the user selected with the noise type the user selected applied to the image. The second image is the image with noise run through the filter that the user selected previously. The third image was the original image the user selected without any noise in order for the user to compare the filtered image with the original. The final image is a DFT of the noise image so the user can adjust input values accordingly to better filter the image.

Throughout the project there where many challenges and discoveries made for each group member based on the filters they were working on.

- Band Filters:
  - Jinman Cai worked on Band-Reject Filter and Band-Pass Filter:
    - Discoveries: I found out that the Band-Reject and Band-Pass filter is good if the location of the noise is known. Because you can use the input to determine the scale and range to tackle that noise. Otherwise you have to play around with the range in order to find and remove the noise.
    - Challenges: I ran into some issues when dealing with the Gaussian band-reject and butterworth Band-Reject filter. It was giving me the errors whenever the denominator is dividing by zero due to the input. To get around this, I have the filter return one if the distance or the denominator is equal to zero otherwise perform the operation.
  - Austin Metcalf worked on Band-Pass/Reject Filters, Notch-Pass/Reject Filters:
    - Discoveries:
      - Discovered band reject filters are decent at removing noise when the location of the noise is well known. The noise can be found by closely analyzing the DFT. However, all of the band filters create a glow/outline effect on objects in the image as well as removing noise. The Gaussian filter, however, had the least noticeable glow effect. Later I discovered that notch filters tend to target noise better than band filters, and learned to implement them. These filters do not have the undesired glow effect.
    - Challenges:

- Had some issues with division by zero in the Gaussian and Butterworth versions of the filters. Also had issues trying to get the filters to target the noise, without distorting the image too much. Had some trouble integrating the filters into the gui and getting the functions to receive the correct input values. Had to update gui to accept more values in order to accommodate notch filters. This was challenging to work into the existing framework.
-

- Order Statistic Filters:
  - Nicolas Graves worked on Min, Max, Adaptive Mean, and Mean Filters:
    - Discoveries:
      - The max, min and midpoint filters where effective at removing salt and pepper noise. Max was very effective at removing salt. Min was very effective at removing pepper. Midpoint was effective at removing both.
    - Challenges:
      - I found a roadblock when trying to implicate min and max filters based on the window size. I lowered the size of the scanning matrix and found an acceptable size. This solution was then changed in place of user input for scanning matrix size.
      - Had some difficulties with my python IDE. My IDE would not run our code so I would work with someone else to ensure my code was working.
  - Joshua Krchnak worked on Alpha Trimmed Mean filter:
    - Discoveries:
      - The alpha trimmed mean filter was good at removing uniform noise such as Gaussian noise. It achieved this by looking at a set number of elements, in this case pixels, within a certain window. It then discards the lowest and largest values then takes the average value and sets that value to be the pixel value.
    - Challenges:
      - Working with a sliding window that orders the pixel values inside from smallest to largest. Had some difficulty arranging the values correctly then discarding the smallest and largest ones.
  - Zachary Brewer worked on Median Filter and Adaptive Mean Filters:
    - Discoveries:
      - The median filter was very effective at removing salt noise, pepper noise, and salt and pepper noise. Also discovered how important it is to find a balanced window size. A large window would be sure to remove all the noise, however, the larger the window the more it distorted the image. It was interesting trying to create a window large enough to remove all the noise, but small enough that you retained as much definition of the original image as possible.
    - Challenges:
      - Ran into problems when trying to create a window size based on the user input. For example, if the user wanted a window size of 4

all the algorithm needed to do was divide 4 in half and get the neighboring pixels from i-2 to i+2 so that there is a 4x4 box around i. The problem arose if the user entered an odd number since they can’t be divided in half. I got around this by dividing the number in half and then taking the ceiling and floor values of the answer.

- Mean Filters:
  - Okeith Percy worked on Adaptive Mean Filters- Arithmetic:
    - Discoveries:
      - When working with the Arithmetic Mean Filter operation on an image, I discovered that it removes short tailed noise such as uniform and Gaussian type noise. It was basically taking the mean of the surrounding pixel values depending on the size of the filter. The larger the filtering mask becomes the more predominant the blurring becomes and less high spatial frequency detail that remains in the image.
    - Challenges:
      - Just issues with if my matrix of the image needed padding but after some research, I found a way to retrieve values efficiently from the image matrix and implement my algorithm.
  - Harshesh Shah worked on Adaptive Mean Filters - Geometric & Harmonic:
    - Discoveries:
      - After completing the algorithms, the pepper images did not work so well with Geometric and Harmonic noises. I learned that the bigger the filter I chose for Geometric the better the result was, although the image got more blurred. The bigger the cutoff frequency, the blurrier its gets but removes most of the noise. The Harmonic worked the best with salt noise and theoretically is supposed to work most efficient with clearing the Gaussian Noise than the arithmetic mean filter. I faced some challenges with the removal of Gaussian noise that I will elaborate below.
    - Challenges:
      - The problems I faced were mainly with Geometric mean filter and Harmonic mean filter requiring multiplication of pixel values and division of pixel values respectively, that caused errors when the image pixel value was black = 0. For some odd reason, my Harmonic algorithm adds salt when filtered with Gaussian Noise. I do not understand why that happens since nowhere in my code do, I change the value of a pixel to 255. This is probably the only thing I was not able to figure out. Apart from that the filters work with other noises.

The GUI was created using the Tkinter library to display the images to the user. There were some issues getting the dropdown menus to update in real time once the user had selected a specific type of noise. Also getting the images to display correctly and anchor to the correct position on the window was a slight challenge.

The mean filters user input was attempted however the full implication was not realized so they are now a set value.

The group asked Harshesh Shah and Okeith Percy to integrate the adaptive filters. They had worked on the filters for two weeks and could not get them to function. Through repeated check-ins with the rest of the group they seemed to be on track, but stated to the group the night before the project was due that they could not get the filter to work. So, Zachary Brewer and Nicolas Graves attempted to see if they could help fix the Adaptive filter. The Adaptive filter now does give a functioning return but is unknown if right.