Horia "Ryder" Stancescu
Hstances - 0721385

Suppressing Noise

Introduction

For the assignment the task chosen is noise suppression. The motive behind this form of Image manipulation instead of the others (acuity & contrast) is primarily due to how easy it is to determine whether an image has been improved or not. When improving contrast or sharpness after a certain degree it becomes very difficult to associate whether or not an image has been improved in an objective matter after being processed, this is due to the subjectivity associated with the preference of the viewer interpreting the images. With noise the observations are no longer as subjective, the noise either exists or it does not, thereby a more objective approach can be used when determining the effectiveness of the algorithms via metrics and visual results. For the remainder of this Report I will be discussing the 3 algorithms outlined in the provided python file "imenh_lib.py" along with a quick algorithm I made based off the existing code which is a median filter included in "imenh_lib.py". The code is a small edit from the code in Hybrid median and mode resulting in the median filter. After the filters get explained they are then compared for which algorithm is best based on the metrics provided in this report.

Alpha Trimmed Mean Filter

What is it and how does it work

The alpha trimmed mean filter is a discrete filtering strategy that is based on order statistic filtering. Order statistic filters are a class of nonlinear digital filters noted for their robust smoothing properties [3]. Therefore alpha trim falling under this Order statistic class of filters is also a non-linear smoothing filter, it functions by taking the "mean" or central tendency and applying a smoothing formula to interpolate between the mean and or the median via an intensity value. This value varies between 0% equivalent to the mean, or 100% equivalent to the median. However as is evident with averages extreme outliers will negatively impact the performance of any average taken as such a certain amount of either one or both of the extreme ends is removed this act is referred to as truncating or trimming [1]. By truncating the results the outliers will not factor into the equation and as such not impact the performance of the algorithm. The intensity value and the amount to truncate are dependent upon the sample size used, typically a value of 5% to 25% of the ends are trimmed, an example would be given a sample size of 8 points a 12.5% trim would discard the minimum and maximum value in the sample [4]. It is important to note that this algorithm scales based on the sample chosen and the number of passes done, meaning that multiple passes of a small sample is approximately equal to a single pass of a larger sample, ie two passes of a 3x3 sample is roughly equivalent to a single pass of a 5x5 sample, three passes of a 3x3 is approximately equal to a 7x7 sample and so on [1].

Note: In the python code "imenh_lib.py" the function "enh_alphaTMean", demonstrates how the alpha trimmed mean works. In this code my use of the term intensity is synonymous with the word "alpha" in the program.

Median Filter

What is it and how does it work

Similar to the above algorithm, the median filter does not function too differently. Unlike mean filtering (low pass), in median filtering the neighbouring pixels have more of an impact on the center pixel. This results in far better handling in particular around the edges of an image unlike in smoothing filters [5]. The way median filter works is by for every pixel taking its neighbours and ranking them based on brightness, the median value for this rank becomes the new value for the pixel being sampled. Similar to alpha trim, its important to note that the neighbours are not solely the immediate neighbours, larger neighbourhoods could be used for the sample such as a 5x5 matrix, the only requirement is that for whatever sized matrix there is exactly 1 center node being evaluated. The median filter offers a few advantages over other low pass filtering techniques such as: not being affected by outliers, edge cases are properly handled, and due to minute edge degradation multiple passes can be done without impacting the image [5]. However this filter is not without flaws, it is important to note that the algorithm can be expensive to compute due to the operations constantly being performed. Median filters are also non-linear, which can result in unwanted results when summing filtered images [5].

Truncated Median Filter

What is it and how does it work

This filter also referred to as the mode filter performs similarly to the Median filter described above. The difference is instead of the median, the mode is calculated and used as the intensity value. The reason it is referred to as the truncated median filter is due to the process of truncating the values before calculating the median value. The reason the median value is chosen is due to the likelihood of choosing a recurring number, the way this likelihood is increased is by trimming a certain percentage of values from both ends of a flattened array this will reposition the median to be centered on a value similar if not equal to what the mode would have chosen. Utilizing this approach the mode can be estimated with relatively good accuracy while not reducing speed by enumerating through a sample window for every pixel. The main reason however why a Truncated Median is used instead of a Mode, is that utilizing a mode in small neighbourhoods the mode is poorly defined and can have random results[5]. Aside from this slight modification, the Mode/Truncated Median Filter does not behave differently from the Median filter aside from this truncation step.

Hybrid Median Filter

What is it and how does it work

This is a filter which draws upon the median filter and improves upon it. It is referred to as a three step process that uses two subgroups of typically 5x5 sample matrix sizes which act as a mask [5]. The two groups are constructed in a manner that divides the image into 4 quadrants, one resembling a cross the other an “x”. In this method both subgroups use their mask and calculate their respective medians, the median between the reference pixel, and the two subgroups are then used for the final pixel. This algorithm has all the benefits of median filter with the added bonus of having better image corner preservation handling [5,6].

Metrics

Mean Opinion Score

For this assignment in testing how well the above filters have performed a series of metrics were used to establish exactly how well each algorithm performed. The first metric used was the Mean Opinion score, although humans are known for being subjective and introducing ambiguity it goes without saying that at detecting visual/spatial differences it is far easier to give an individual an image and ask them to analyze and compare than a computer. However it is still important to reduce ambiguity therefore the following steps were taken to attempt this: first by sampling from a size of 10 people and the most common occurrences as the index, this will in theory reduce individual biases. The second task was stating what each numeric value represents when judging how well the algorithm performed, thereby instead of the individuals giving varying results as to what each one believes a “3” is, instead it is stated above the value and the participant simply has to determine based on what they saw, the statement that best relates with the observation. Using this defined rating system, and a large sample size, the results can be used to determine which algorithm performed best at what area with minimal subjectivity involved.

The Questionnaire was structured as such:

		Noise Removal				Image Preservation			
		Almost zero noise was removed	Some noise was removed but quite a lot still exists	A lot of noise was removed but some still exists	Almost all if not All noise was removed	The image looks nothing like the original	The image resembles the original, however it is clearly distorted	The image is close to the original, but it is slightly distorted in some areas.	The image looks exactly as the original
Original Image	Filtered Image	0	1	2	3	0	1	2	3

Although these types of questions may not entirely remove subjectivity, it is impossible to realistically ask an individual if the number of noise ranges between two numbers as that would imply they have to count the noise, and no one is willing to sit around counting noise for a survey considering most people would avoid surveys to begin with, as such by grouping results as to what they believe “some” versus “a lot” is results can be obtained without the increased subjectivity described above referring to the number 3 and its meaning. For the results obtained the metric is not about how each individual scores an image but rather as a collective implying how the average person would score the results.

Structural Similarity Metric

This metric was used to identify how well the above algorithms remove noise while not altering the image in a negative way. Although it’s easier to tell via the eye how altered an image is, for practical purposes it’s often more preferential to have a numeric value represent the image structure itself. Utilizing Structural Similarity an index of -1 to 1 is reported based on how well two image sets match, in the case of a perfect or near perfect match the results will be more inclined to a value of 1. All results from this calculation are stored in the text file labelled “SSIM_results.txt”, in this file every single filtered image is compared against the original to retrieve an SSIM index number, and that value is reported alongside the filtered image.

Ranks Noise Estimation Index

This metric is used to identify the amount of noise removed by determining how much is present. The value returned can vary greatly typically an image with zero to small amounts of noise will hover in the low area of an index of 1 to 0, while the more noise that exist the higher the value, values upwards of 500 can even be reached given enough noise and a large enough image.

Time Estimation

As the performance of algorithms are what’s being determined, it’s only natural that a time stamp metric is also used to determine the runtime length of an algorithm. Whether an algorithm can provide excellent results is one matter, but if it requires a considerable amount of time to reach said results than the algorithm can be deemed as unrealistic and unusable or in need of modifications/speed hacks to improve its runtime. As such this metric is included to provide a greater understanding as to the true “cost” of achieving the results from each filter. Results are stored in “Time_results.txt”.

Metric Results:

As mentioned above the metrics used were Mean Opinion Score from a sample size of 10 to get an average human analysis as to how well each filter performed, Structural similarity Metric to determine the integrity of an image post-filter, and finally the Time Estimation metric to determine the runtime length of each algorithm. For the Mean Opinion Score, the 10 participants were chosen randomly (in that they were the first 10 people to not refuse the survey), the survey took about 5 minutes per person as there were 4 Filters with 11 pictures each = 44 total images that needed scanning = 44 separate scores. The averaged scores are stored in a CSV file named “MOS_perPerson.csv” included with this report, note the values are cutoff in the case of decimals to have whole numbers.

Based on the MOS metric all result averages are stored for each image and filter as true values under the corresponding column simply analyzing the number of occurrences reveals patterns along with feedback from the participants. The main consensus is in regards to noise all algorithms did a phenomenal job at removing the noise in most cases. However in regards to original image preservation there was a pattern that emerged, the pattern was each filter had its own trademark signature in comparison to the others, hybrid median on average resulted in the same image with negligible distortion, truncated median resulted in water-paint renditions of the original image, alpha trimmed resulted in often times what appeard to be faded representations of the original image, and finally median resulted in a slight image blur.

There were a few extreme photos which were placed mainly to see how the filters react as opposed to the metrics which had interesting results. The image "grayscale_4_shades-600x375" which is a grayscale image of a cartoon drawing from an old favourite video game of mine "Escape from MonkeyIsland resulted in a complete abstract and almost chromatic result for all four algorithms, ironically when fed water-paint like art, the truncated median filter had slightly less distortion than the others. The next image is "halftone_noise" in the original it's a picture of a child made up of black and white noise, the noise actually makes the picture, feeding this into all 4 filters results in a complete loss of image and failure in even noise removal. (Questionairesample.pdf)

The SSIM metric gave a numeric value to degree to which images get distorted. Overall it would appear that Hybrid median filter had the lowest amount of image degradation based on its consistency of having a higher value in regards to the other filters, the median filter has a tendency of coming in second or even first on some occasions, truncated mean averaging almost always subpar to median, and alpha trim consistently offering the worst structural integrity of the four algorithms (SSIM_results.txt).

Ranks Noise had some very interesting results. The algorithm determined that in most cases all 4 algorithms did a perfect job at removing noise, which most people from the MOS would agree with. However its findings are anything but consistent, on some images a value of "inf" was provided, and on the two "rigged" images "halftone_noise" and "grayscale_4_shades" a value of "nan" was retrieved. However on the images where metrics were uncovered in regards to rank it would appear alpha trim was by far the worst at noise removal, followed by median, and then truncated median, with hybrid being the best always attaining a lower number than the others. (RankNAI_results.txt)

The last metric time, gives a nice indicator as to what algorithm would typically be chosen based on runtime. Typically on average hybrid median and truncated clocked in the most time as noted by noise_1 with Hybrid median requiring 35seconds, truncated taking 36, median taking 12, and alpha taking 10. This tendency tends to continue throughout all 12 images, with truncated being the most expensive, hybrid second, median third, and the cheapest being alpha trimmed mean. (Time_results.txt)

My conclusion would be that with the metrics given, and the visual results of the images, that truncated median should never be used if image preservation is the goal as the distortion is large, and the computation time is the largest one meaning it is the obvious worst choice to choose leaving only the other three filters. Alpha trimmed I would recommend when manipulating large images due to its speed even though its low scores across the other metrics, median for images which are slightly blurred to begin with, and finally hybrid median when best results are needed. This is due to although hybrid median is the second most computationally expensive, it was the best at structural integrity and noise removal, thus the obvious best choice in regards to quality.

References:

1) http://subsurfwiki.org/wiki/Trimmed_mean_filter

2) http://labsenales.uniandes.edu.co/index_archivos/adaptive%20trimmed.pdf

3) J. W. Tukey, "Nonlinear (nonsuperposable) methods for smoothing data," in Conf. Rec., 1974 EASCON. p. 673

4) http://en.wikipedia.org/wiki/Truncated_mean

5) http://medim.sth.kth.se/6l2872/F/F7-1.pdf

6)http://www.librow.com/articles/article-8

7)] Rank, K., Landl, M., Unbehauen, R., “Estimation of image noise variance”, IEE Proceedings Vision, Image and Signal Processing, 146, pp.80–84 (1999)