Assignment 1

# Processing grayscale images!

Erik Zorn-Wallentin

0864583

Due: Feb. 3rd

CIS \* 4720

Professor: Michael Wirth

# Table of Contents

## Table of contents for Assignment 1

**i Part 1: Bad Photos**..……………………………………………………………………..1

**ii Part 2: Picture Processing**..……………………………………………………………2

**iii References**..………………………………………………………………………………3

# i Part 1: Bad Photos

I found a series of photographs online that illustrate artifacts on each of the images. These 6 type of images may be over overexposed, or underexposed, or blurry, or incorrect white balance, or noise. I attempted to remove or suppress the artifacts in each of the images where I also gave the histogram, an explanation of defect and the results with the enhanced image.

|  |  |  |  |
| --- | --- | --- | --- |
| **Original Image** | **Histogram and Brief explanation of defect** | **Statement on if image defect can be suppressed/removed** | **Enhanced Image** |
| C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\Part 1\overexposed1.png | C:\Users\Erik\Desktop\histogram_1.png  Image is overexposed, can be fixed by stretching out the histogram. | I applied Enhanced Contrast and this was the best result I could get, so the overexposed image can barely be suppressed and arguably can’t be fixed. | C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\overexposed1_enhanced.jpg |
| C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\Part 1\overexposed2.jpg | C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\Part 1\histogram_2.png  Image is overexposed, can be fixed by stretching out the histogram. | I applied Enhanced Contrast. As you can see the overexposure of image can be suppressed on the overall parts of the image using Enhanced Contrast. The primary part not so much but that is intended. | C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\overexposed2_enhanced.jpg |
| **Original Image** | **Histogram and Brief explanation of defect** | **Statement on if image defect can be suppressed/removed** | **Enhanced Image** |
| C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\Part 1\underexposed-6-72dpi-1500px-web.jpg | C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\Part 1\histogram_3.png  Image is underexposed, could be fixed using Enhanced Contrast. | I applied Enhanced Contrast. As you can see the underexposure of image can be suppressed using Enhanced Contrast. | C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\underexposed1_enhanced.jpg |
| C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\Part 1\noisy.jpg | C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\Part 1\histogram_4.png  Image is noisy, could be fixed using Smoothing. | I applied Smoothing. As you can see the noisy image can be suppressed using smoothing. | C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\noisy_enhanced.jpg |
| C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\Part 1\blur and out-of-focus.jpg | C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\Part 1\histogram_5.png  Image is blurry due to motion and out-of-focus. | I applied Sharpen and this was the best result I could get, so the blurry and out-of-focus image can’t be fixed as everything is still out of focus and blurry. | C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\blur and out-of-focus_enhanced.jpg |
| C:\Users\Erik\Desktop\incorrectWhiteBalance.png | C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\Part 1\incorrectWhiteBalance_histogram.png  Image has incorrect white balance as it has a blue tint to the image which needs to be removed. | Applied White Balance technique and the blue tint of the image went away and was able to fully remove the incorrect white balance. | C:\Users\Erik\Desktop\incorrectWhiteBalanceFixed.png |

# ii Part 2: Picture Processing

The topic I decided to choose is Suppressing Noise. A lot of older images contain noise, and I was always curious growing up on how to remove noise from images and I decided this was a good time to use images with noise and try to eliminate the noise in them. Noise is an artifact in an image, which is a random variation in image density. An example of noise would be film grain noise, where you can visibly see “grain” in the film and pixel level variations in images.

**Algorithm Discussion**

I will attempt noise suppression techniques on noisy images to reduce the noise in those images. Some noise suppression techniques out there are median filtering, mode filtering, alpha-trimmed mean filter, Kuwahara filter, Nagao-Matsuyama filter, Wiener filter, centre weighted median filter. I will use the metric such as the estimation of noise variance described by Rank, in order to decide how well the noise suppression algorithms have worked.

The noise suppression techniques I used on the provided images were (from provided library in Course Link under python code called “imenh\_lib.py”) alpha-trimmed mean filtering, truncated median “mode” filtering, and hybrid median filtering. The noise suppression techniques I used from my own implementation and the two that I did were, Gaussian filtering and rank filtering.

The alpha-trimmed mean filtering uses an alpha value(0.5 in this example) where it describes how many elements to trim from the sorted vector representing the neighbourhood image. The truncated median “mode” filtering contains characteristics of the median, truncated median and mode filters. It also uses an alpha value of 5. The hybrid median filtering extracts the median from a 5x5 neighbourhood using two masks (5x5 grid). After the hybrid median will be calculated as the median of the three values of the two medians from the 5x5 masks.

For my own algorithms Gaussian filtering more of the weight is given to central pixels than those outside. I chose an sigma of 1 in my implementation. Also for rank filtering it is a non-linear filter which can also be called “order statistic”, it compiles a list of intensity values in the neighbourhood of a given pixel, puts it into a list from ascending order than selects a value from a certain position in the list to use as the newest value for each pixel.

On the next page you will see I performed experiments on the provided 4 images in Course Link of this class from “Set 1 Noisy Images”. I performed the 5 (2 my own) noise suppression techniques as described above on each of the images and gave the results. Before performing the suppression technique I did lower the resolution on the base images for some of them to get quicker results. I also converted one of the .tif images to .jpg to stay consistent with everything. I also found 2 images online and also did the same suppression techniques on them to be included in the experiments.

**Experiment Design: Library provided.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Original** | **Alpha-trimmed mean filtering** | **Truncated median “mode” filtering** | **Hybrid median filtering** |
| C:\Users\Erik\Pictures\n1_original.png | C:\Users\Erik\Pictures\n1_alpha_fixed.png | C:\Users\Erik\Pictures\n1_truncated_fixed.png | C:\Users\Erik\Pictures\n1_hybrid_fixed.png |
| C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\Part 2\n2_original_fixed.png | C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\Part 2\n2_alpha_fixed.png | C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\Part 2\n2_truncated_fixed.png | C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\Part 2\n2_hybrid_fixed.png |
| **Original** | **Alpha-trimmed mean filtering** | **Truncated median “mode” filtering** | **Hybrid median filtering** |
| C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\Part 2\n3_original_fixed.png | C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\Part 2\n3_alpha_fixed.png | C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\Part 2\n3_truncated_fixed.png | C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\Part 2\n3_hybrid_fixed.png |
| C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\Part 2\n4_original.png | C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\Part 2\n4_alpha.png | C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\Part 2\n4_truncated.png | Crashes the algorithm, unable to get enhanced image because of the crashing of algorithm on this image. Zoomed in version of image was only change. |
| C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\Part 2\n5.jpg | C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\Part 2\n5_alpha.png | C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\Part 2\n5_truncated.png | C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\Part 2\n5_hybrid.png |
| **Original** | **Alpha-trimmed mean filtering** | **Truncated median “mode” filtering** | **Hybrid median filtering** |
| C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\Part 2\n6.jpg | C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\Part 2\n6_alpha.png | C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\Part 2\n6_truncated.png | Crashes the algorithm, unable to get enhanced image because of the crashing of algorithm on this image. No changes were made from online source. |

**Experiment Design: My Two Algorithms**

|  |  |  |
| --- | --- | --- |
| **Original** | **Gaussian filtering** | **Rank filtering** |
| C:\Users\Erik\Pictures\n1_original.png | C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\A1ezornwalCIS4720\n1_gaussian.png | C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\A1ezornwalCIS4720\n1_rank.png |
| C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\Part 2\n2_original_fixed.png | C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\A1ezornwalCIS4720\n2_gaussian.png | C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\A1ezornwalCIS4720\n2_rank.png |
| C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\Part 2\n3_original_fixed.png | C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\A1ezornwalCIS4720\n3_gaussian.png | C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\A1ezornwalCIS4720\n3_rank.png |
| C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\Part 2\n4_original.png | C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\A1ezornwalCIS4720\n4_gaussian.png | C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\A1ezornwalCIS4720\n4_rank.png |
| C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\Part 2\n5.jpg | C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\A1ezornwalCIS4720\n5_gaussian.png | C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\A1ezornwalCIS4720\n5_rank.png |
| C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\Part 2\n6.jpg | C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\A1ezornwalCIS4720\n6_gaussian.png  Appears to be something wrong with source file. But I decided not to make any changes to show the results in this scenario. | Crashes the algorithm, appears to be something wrong with the source file. But I decided to not make any changes to show the results in this scenario. |

As a side note I did zoom in on some of the images that were too large, which caused the weird results in the fourth image as it doesn’t happen when I don’t zoom in but I left it in to show what happens. No changes were made to sixth image or seventh image but also gave interesting results.

Based on my results, the 5 (2 my own) suppression techniques used on all 6 images showed the different results based on the techniques. For the first, second, third, and fifth image and applying all 5 techniques in my opinion the final result of hybrid median filtering looks the best visually. A very important note on results, the hybrid median filtering algorithm crashed in a scenario, and rank filtering crashed in a scenario. I decided to not modify the library and algorithm provided from Course Link python files and left it as is. This could be a flaw in the algorithm or in the images provided where it could not do the filter on them based on the specific files. Also the fourth and sixth image looks “inverted” and also now very overexposed and underexposed which is very incorrect comparing to the original image. Under further testing I believe the size of the image has an impact on this.

**Testing**

The images provided vary in size, from 751x415 resolution up to 3344x2052. Naturally this is a huge difference in resolution and would drastically make a difference on how long it would take to generate the enhanced images using suppression techniques. I provided below the testing times of each algorithms based on resolutions provided. I used test1.jpg as 751x415 and I up-scaled it to 334x2052.

Time taken for alpha-trimmed median filtering:

|  |  |  |
| --- | --- | --- |
|  | **First Image(test1.jpg)** | **Second Image (test2.jpg)** |
| **Resolution of Image** | 751x415 | 3344x2052 |
| **Time Taken** | 2.66 seconds | 67.71 seconds |

Time taken for truncated median “mode” filtering:

|  |  |  |
| --- | --- | --- |
|  | **First Image(test1.jpg)** | **Second Image (test2.jpg)** |
| **Resolution of Image** | 751x415 | 3344x2052 |
| **Time Taken** | 13.22 seconds | 309.85 seconds |

Time taken for hybrid median filtering:

|  |  |  |
| --- | --- | --- |
|  | **First Image(test1.jpg)** | **Second Image (test2.jpg)** |
| **Resolution of Image** | 751x415 | 3344x2052 |
| **Time Taken** | 15.01 seconds | Crashed |

Time taken for Gaussian filtering:

|  |  |  |
| --- | --- | --- |
|  | **First Image(test1.jpg)** | **Second Image (test2.jpg)** |
| **Resolution of Image** | 751x415 | 3344x2052 |
| **Time Taken** | 0.006 seconds | 0.88 seconds |

Time taken for rank filtering:

|  |  |  |
| --- | --- | --- |
|  | **First Image(test1.jpg)** | **Second Image (test2.jpg)** |
| **Resolution of Image** | 751x415 | 3344x2052 |
| **Time Taken** | 0.009 seconds | Crashed from array type |

Following the tests above, while using the two same images with drastically different resolutions (one smaller vs much bigger) appears to have drastically different results on each algorithm. For one the Gaussian filtering algorithm was the quickest out of all the algorithms used. Also the second image (bigger one) drastically took much longer than the smaller one in each of the algorithms. An important note, the hybrid median filtering timing results shows to crash on the larger image which also matches the above output on why I couldn’t get the fourth and sixth image results on hybrid median filtering as the original image is very large.

**Difficulties**

There were a few difficulties in processing images within the noise suppression topic. Using the 3 algorithms, they all had different times on how long it took to calculate the same size image. Where an image of the size 751x415 took 2.66 seconds for alpha-trimmed median filtering, while the same image took 15.01 seconds for hybrid median filtering. Just using different algorithms is difficult if the algorithm itself takes too long to calculate the enhanced version of the image. Also another difficultly in processing images was the large resolution image (3344x2052) takes much longer to calculate in each of the algorithms compared to the smaller resolutions (751x415). The difference from smaller image taking 2.66 seconds in alpha-trimmed median filtering to 67.71 seconds for the larger resolution is a huge difference and a very important difficulty in processing the images for different algorithms. Another difficulty that I had with the given python library from Course Link called “imenh\_lib.py” (I did not change the source code) is image called “schwyz\_townhall.tif” in Set 2 Noisy Images when modified to jpg and tried to lower the resolution would create new artifacts in the enhanced image using the same algorithms to the point where the new enhanced image is an overexposed and underexposed result which is very incorrect.

Here is an example of that problem

|  |  |
| --- | --- |
| C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\Part 2\n4_original.png | C:\Users\Erik\Desktop\CIS 4720\Assignments\A1\Part 2\n4_alpha.png |

# iii References

Part 1 References

https://abdullahnazib.files.wordpress.com/2016/10/noisy1.jpg?w=274&h=274

https://s-media-cache-ak0.pinimg.com/236x/d8/3c/6a/d83c6a11aff1c7313d73d2a0b563c220.jpg

https://s-media-cache-ak0.pinimg.com/736x/62/7f/a8/627fa8a8c99a2692485f568ba446847f.jpg

https://www.pixoto.com/images-photography/animals---dogs/portraits/overexposed-monochrome-of-black-dog-in-snow-4524005646139392.jpg

https://c1.staticflickr.com/8/7010/6466872285\_12e18fc677\_b.jpg

https://farm4.static.flickr.com/3664/3363677584\_e73fdea495\_b.jpg

<https://static1.squarespace.com/static/57bbd3be2994ca8e30d695c8/57ef8dcf5016e13a6d3e8791/57ef8e3e2e69cf154bb942ca/1476313090816/underexposed-6-72dpi-1500px-web.jpg>

<https://i.ytimg.com/vi/MUh3vZT0tbo/hqdefault.jpg>

Part 2 References

<https://courselink.uoguelph.ca/d2l/le/content/454932/viewContent/1537293/View>

Function for image enhancement (e.g. noise reduction) library “imenh\_lib.py” provided from courselink used as noise suppression techniques.

<http://snpngtrickz.blogspot.ca/2014/02/monochrome-photography-make-striking.html>