DIP Fall 2024 #58195
HW3
Jeffrey Kerley
17 October 2024

Noise Filtering

Abstract:

Digital image pixel noise is a horrible problem for many applications in image processing. Sensor issues, transmission errors, or just general algorithm problems can result in pixel noise scattered throughout an image at varying degrees. A common form of this noise is Guassian noise, where each noisy pixel is independent and uncorrelated from other noisy pixels, but it follows a normal distribution across the image for some mean and SD. As a result, filters have been created to minimize noise and find the original images. Using mean squared error as a metric to track similarity between images, a median, gaussian, and adaptive median filter are examined in their effect to reduce noise on images.

Introduction:

The extent of noise in signal processing is extreme, but it can be understood. In fact, the kinematic description of atoms follow gaussian statistical patterns, so it is no surprise noise is introduced into sensors and other telecommunication pathways following the same gaussian distribution (an approximation…). Taking this idea to heart gives ways to reduce this noise in images following a statistical approach. Gaussian filter, Median filter, and Adaptive Median filter are all reliable approaches to 'undo' the gaussian pixel noise to differing degrees of success. Using MSE as an image similarity metric has its issues, but when tracking changes between a noisy image and the non-noisy ground truth image, the metric works effectively as a metric to track how 'good' a denoising filter is working.

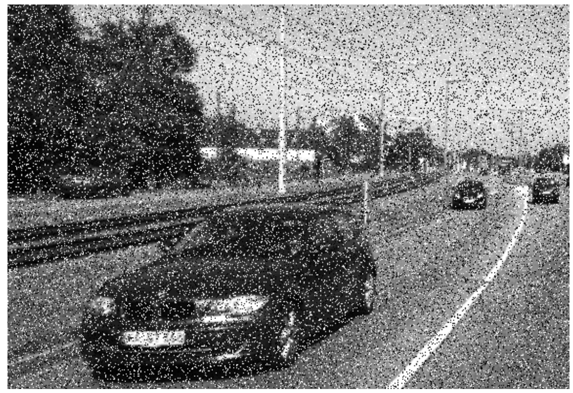
Gaussian filter is a weighted convolution using weighted values according to a normal distribution (mean and SD pixel scaling). This has the effect of blurring the image, while removing noise since they will get averaged out by the normal distribution of values. Median filter does a convolution but replaces the current pixel with the median pixel value from the neighborhood. This will remove noise very effectively, since noise values will never be the median, unless the distribution of the noise in the image becomes large. Finally, the adaptive median filter applies a normal median filter, but the size of the kernel grows to handle cases where there is a large amount of noise in the image. The kernel grows (until SMAX value), until this conditional is satisfied: ***zMin < zMed < zMax.***

Where, **zMin = min of current window, zMed = median of current window, zMax = max of current window.** This conditional attempts to handle cases of many noisy pixels, since we want to keep growing to get the kernel until our median is a non noisy pixel. If the median is the same value as the max, then we ~know the median is a noisy pixel and get more of the image to compensate.

This works the same way for non-white pixel noise values, which is why we also check the min pixel value being less than the median value. This is taken a step further, but the adaptive filter also will ‘reject’ or ‘accept’ pixels if they correspond to noisy vs non-noise pixel. For, ***zMin < zCurrent < zMax,*** we have already confirmed our window pixels are good (median is representative), then we make the claim that the **zMin** and **zMax** are our ‘noise classes’ which means the current pixel (**zCurrent**) should not be changed. In the opposite case, we accept the median pixel, because this must imply that our current pixel belongs to one of the ‘noise classes’ and does need to be changed. Now, results of differing filter types and sizes are applied on images with low and high amounts of noise.

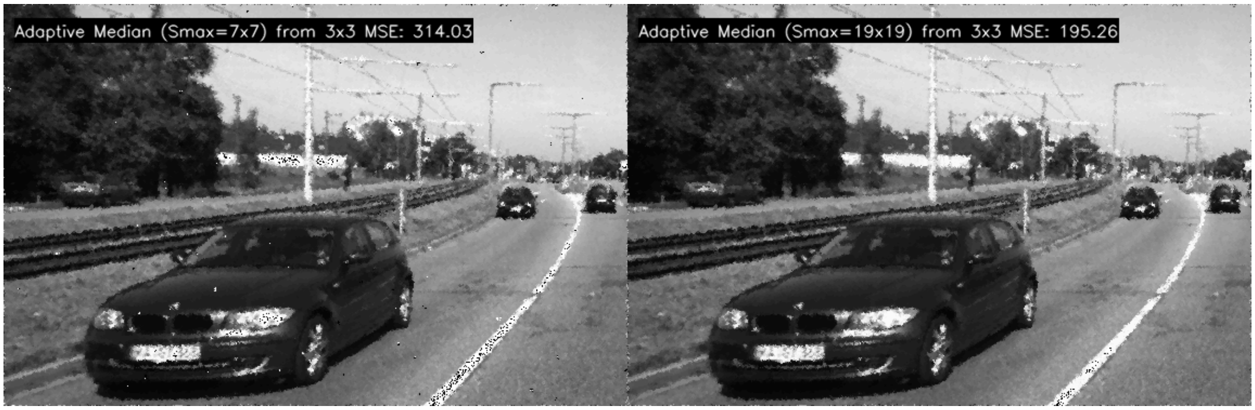
<u>Experiments and Results:</u>

Test1Noise1 (Gaussian and Median Filter: MSE with Test1 = 4357.1337890625):

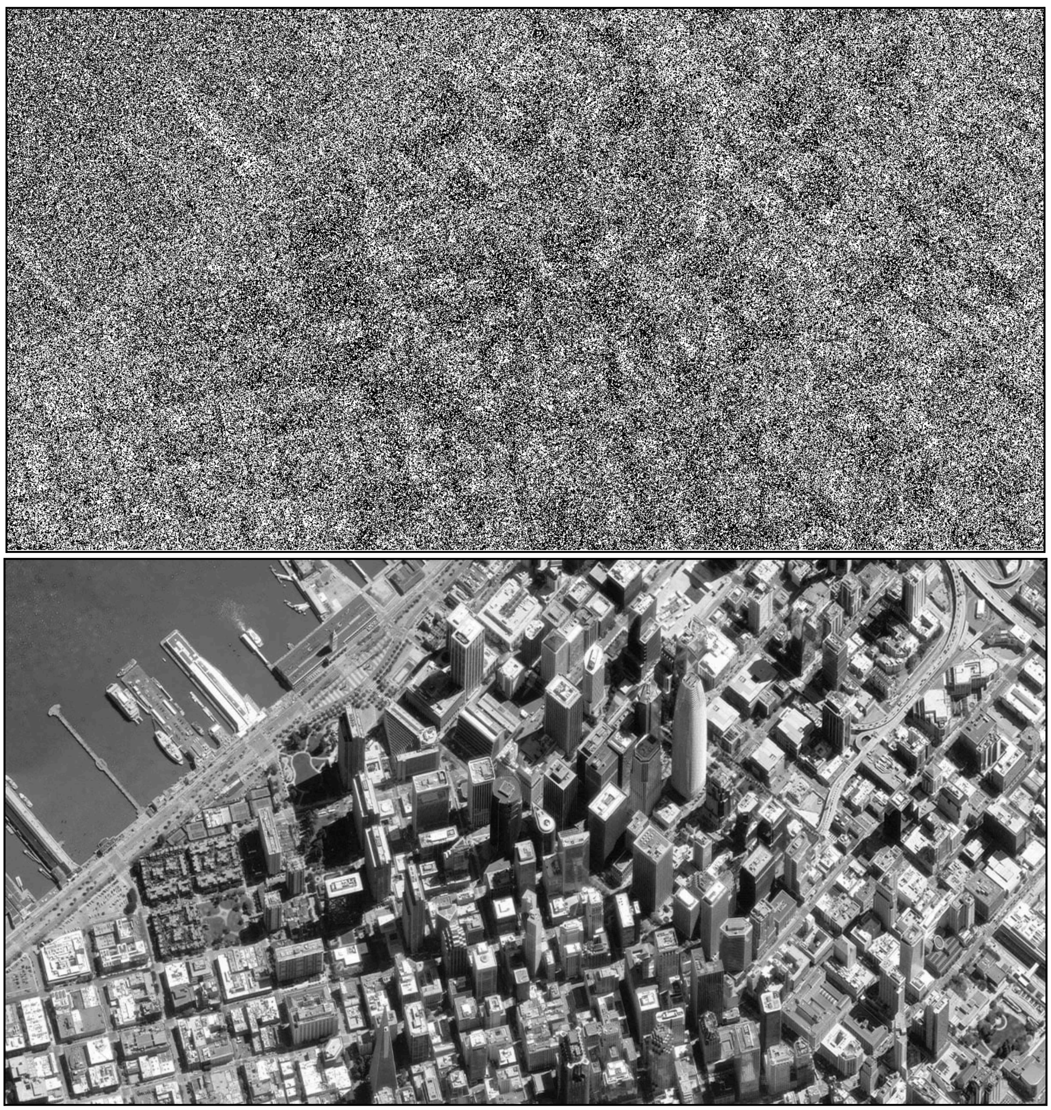


Test1Noise2 (Guassian and Median Filter: MSE with Test1 = 12983.4287109375):

Test1Noise2 (Adaptive Median Filter: SMAX 7x7 = 4.86s, SMAX 19x19 = 4.95s):



Test2Noise2 (Adaptive Median Filter: SMAX 7x7 = 20.01s, SMAX 19x19 = 20.19s):




Adaptive Median (Smax=7x7) from 3x3 MSE: 994.22


Adaptive Median (Smax=19x19) from 3x3 MSE: 777.94

Discussion/Conclusions:

The results had the adaptive median filter working very well! No surprise, but every filter worked to improve the quality of the image (removing noise). The gaussian filter and median filter worked well at low noise levels, which is what was predicted due to the nature of averages. As noise increased, gaussian and median failed harder because the amount of noise became the average. The adaptive median filter uses the idea that salt and pepper noise will be the min and max, and so any pixel that falls between these two should be kept the same. If it does not, then it must be noise. Thus, at high noise amounts this filter removes almost all of the noise while giving a very close approximation of the original image (MSE values similar to normal median filter on low levels of noise). This same idea of growing the filter size does not work with the normal median filter, since as the kernel grows, there is no further check if the data in the current kernel is actually representative of the current pixel to be able to use the median.

References/Appendix:

Plotting: https://www.geeksforgeeks.org/python-opencv-cv2-rectangle-method/

OpenCV doc plotting: https://docs.opencv.org/4.x/dc/da5/tutorial_py_drawing_functions.html

OpenCV doc filters:

https://docs.opencv.org/4.x/dc/dd3/tutorial_gausian_median_blur_bilateral_filter.html