Homework 3: Noise Filtering

Zach Brown - Oct. 03, 2023 - DIP 4650

Abstract

This homework focused on the efficient implementation of the Adaptive Median Filter (AMF), as well as familiarizing ourselves with the Mean Squared Error (MSE) algorithm and OpenCV filtering functions. The results obtained lead to the conclusion that this adaptive mean filter is effective at removing noise from an image in a timely manner.

Introduction

The first task was to implement the MSE algorithm for two images of the same shape. The second task was to implement a few different OpenCV filtering functions, namely the Gaussian Filter and Median Filter, and checking the results with the previously implemented MSE algorithm. The third and final task was to implement the AMF algorithm in an efficient manner, and record various statistics about our implementation. The approach I took with implementing the AMF algorithm was slightly different from the one shown in class, as I inlined the Stage A and Stage B sections. I did this because there is no reason to keep them separate but for an understanding of the logic behind the algorithm.

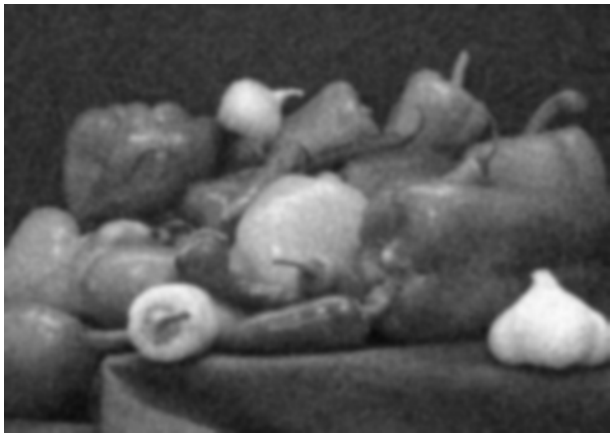
Experiments

Built-in Noise Filtering Images

Test1Noise1

Original

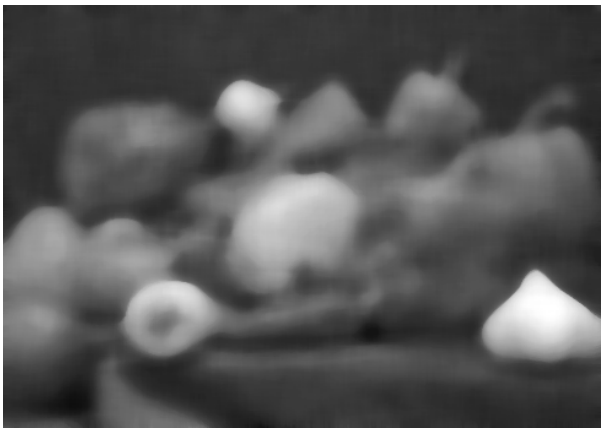
Noisy



Gaussian (sigma = 2)

Gaussian (sigma = 7)

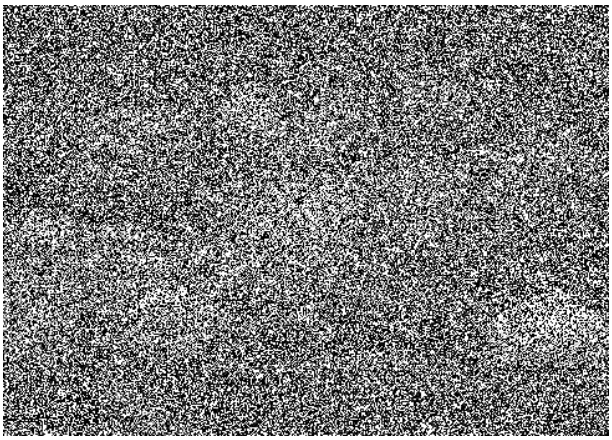
Median (kernel = 7)



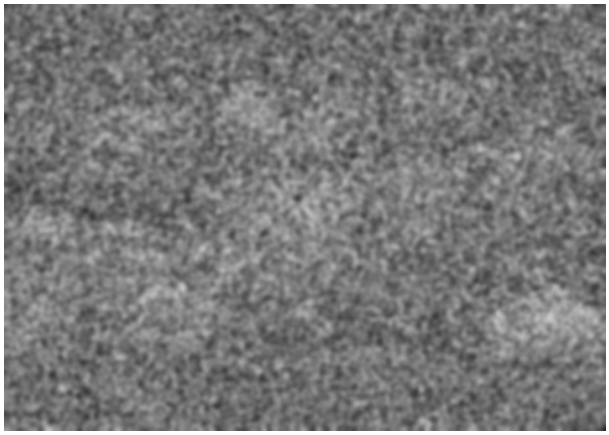
Median (kernel = 19)

Test1Noise2

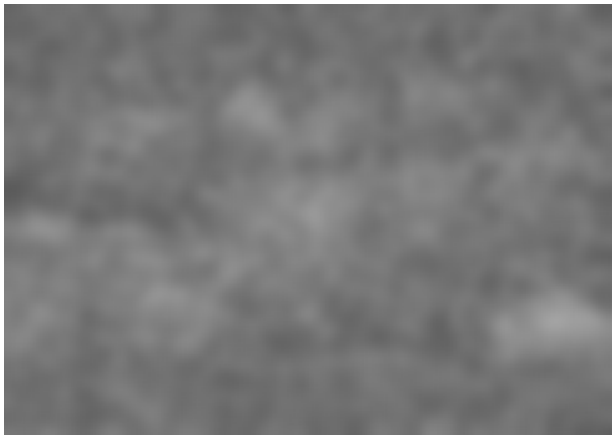
Original



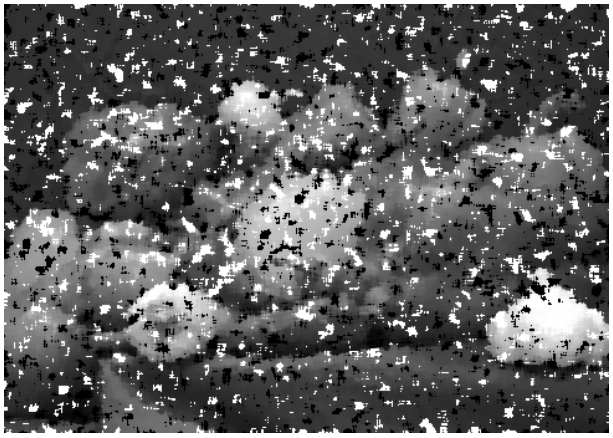
Noisy



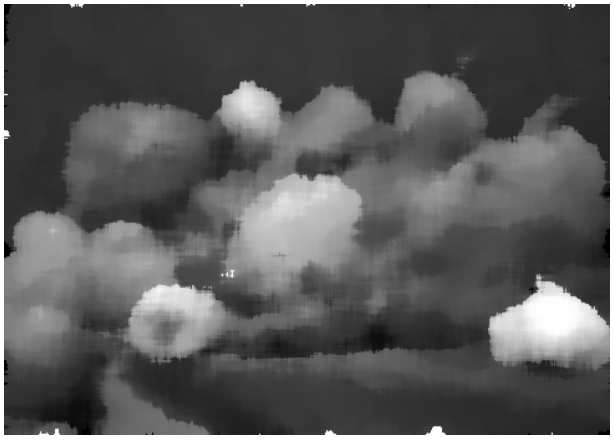
Gaussian (sigma = 2)



Gaussian (sigma = 7)



Median (kernel = 7)



Median (kernel = 19)

Measured Results of Built-in filters

	MSE				
	Noisy	Gaussian Filter sigma=2	Gaussian Filter sigma=7	Median Filter kernel=7	Median Filter kernel=19
Test1Noise1	621.11	66.38	281.42	67.96	189.09
Test1Noise2	16182.17	2881.84	2751.70	3223.11	334.88

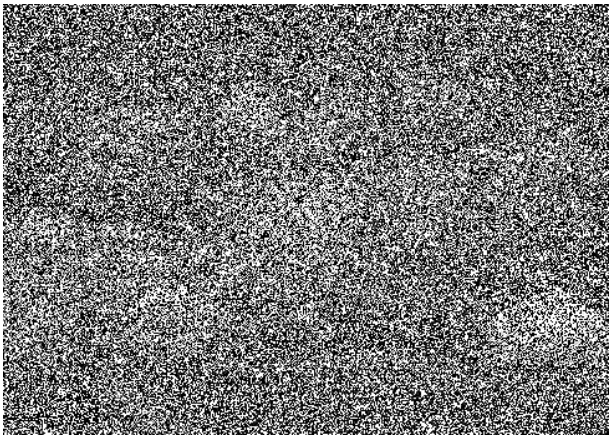
Discussion of Results of Built-in filters

While both filters did a great job in Test1Noise1 when noise was minimal, the only filter-parameter set that had any kind of success on Test1Noise2 was Median Filter at kernel size = 19 pixels. The gaussian filter wasn't able to filter out the salt-and-pepper noise from the image, just resolving into a blurry mess. The median filter was only able to filter out so much noise when the kernel was sufficiently large.

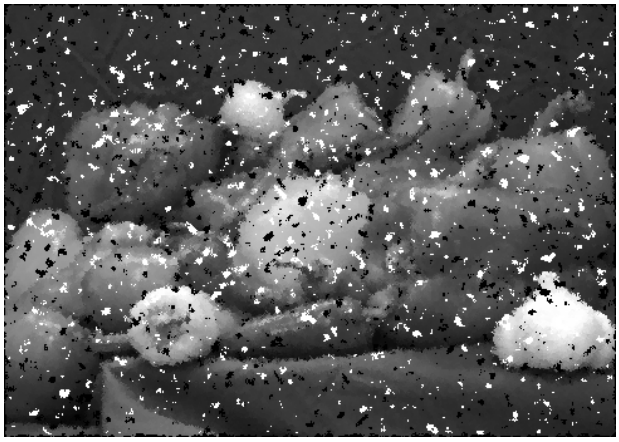
Adaptive Median Filtering Images

Test1

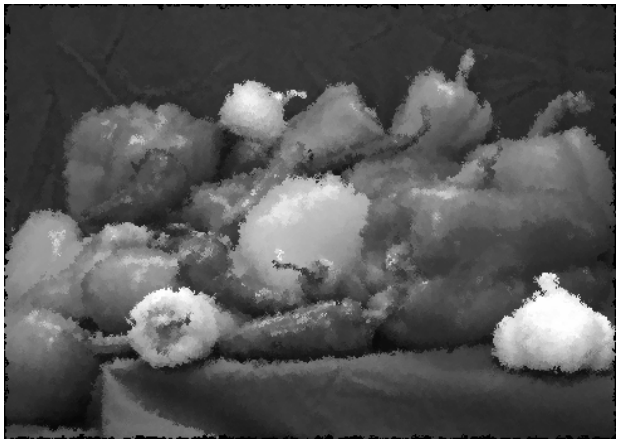
Original



Noisy



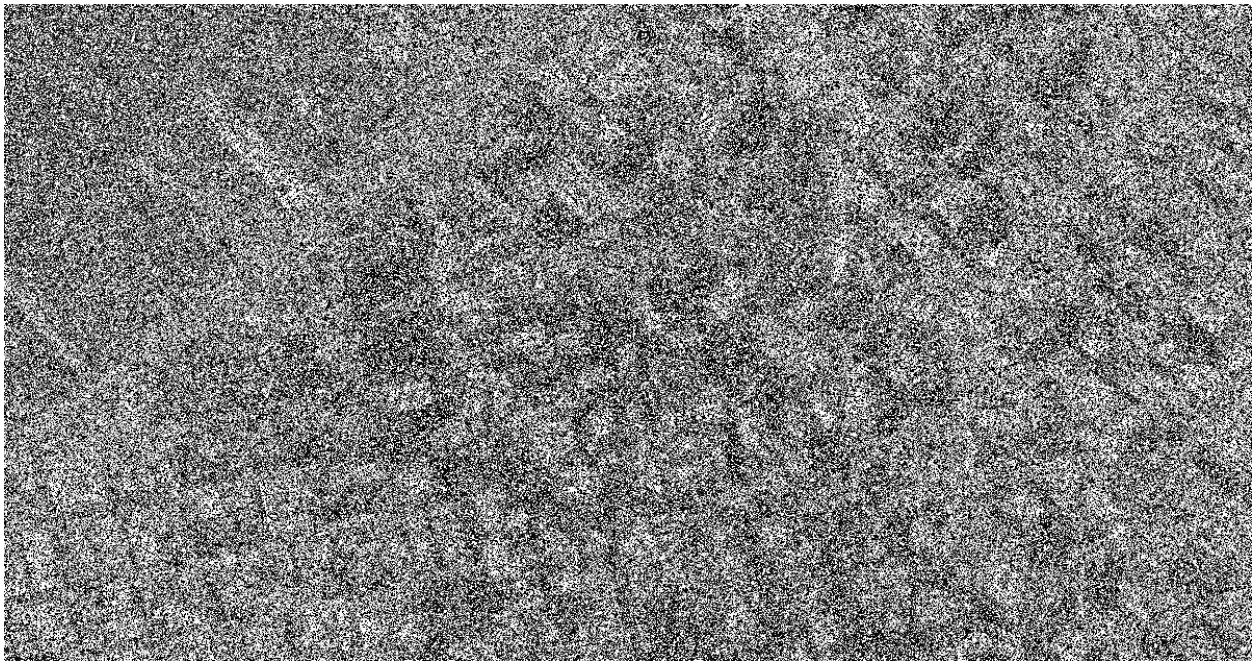
S_{max} = [7x7]



S_{max} = [19x19]

Test2

Original



Noisy

S_{max} = [7x7]

S_{max} = [19x19]

Measured Results of Adaptive Median Filter

	MSE			Processing Time	
	Noisy	Adaptive Median Smax=[7x7]	Adaptive Median Smax=[19x19]	Adaptive Median Smax=[7x7]	Adaptive Median Smax=[19x19]
Test1Noise2	16182.17	1663.67	193.09	11.409s	12.731s
Test2Noise2	14175.33	1110.91	851.98	43.068s	44.636s

Discussion of Results of Adaptive Median Filter

While the MSE of these images may not be as impressive as some other filtering methods, the edges are well preserved and the images are clearly interpretable. The images for Test1Noise2 are more impressive in both MSE and in edge quality than all built-in filters were able to produce. With this much noise, the larger kernel size is very impactful for image quality while only having a small hindrance on performance.

Conclusions

In conclusion, my implementation accomplishes all of the assigned tasks in a timely and efficient manner. While the resulting images of the AMF algorithm weren't restored as well as some of the examples shown in class, I do think that it does a good job of removing large amounts of noise while preserving edges to the best of its ability.