Sabancı University, FENS
CS419 Digital Image and Video Analysis
Assignment 3

Filiz İpek Oktay - 30780

November 2024

1 Morphological Hole Filling

In this task, morphological hole filling has been implemented and applied onto the original image given below. Based on lecture slide logic (P. Soille), the functions use reconstruction by erosion to fill enclosed regions in images. A marker image initializes boundaries, and iterative erosion fills holes while preserving object edges. The process halts when no changes occur, effectively filling regions without altering object boundaries, aiding segmentation preprocessing.

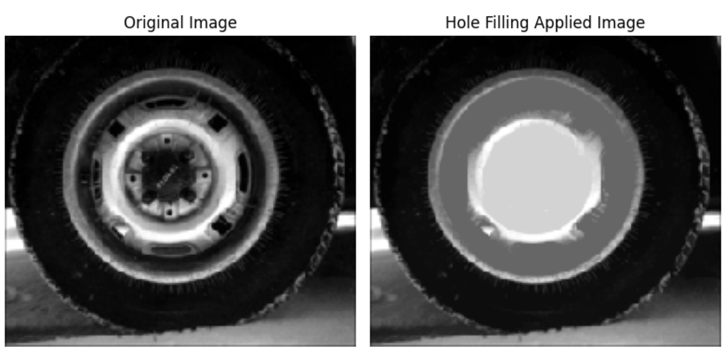

Figure 1: Hole Filling Applied Image

2 Denoising the Sinusoidal Noise from the Corrupted Image

The Fourier Transform is a powerful tool for analyzing the frequency components of an image, particularly useful for addressing sinusoidal noise in images such as the provided aerial photograph of Pompeii. It is important to consider the lack of time efficiency of the manual Fourier Transform (FFT) implementations. Since it was taking too much time to conduct the FFT's along the functions, the OpenCv's functions have been used.

Initially, the 2D Fourier Transform of the corrupted image is computed and shifted to center the low-frequency components, producing a magnitude spectrum that reveals the noise as distinct peaks in the frequency domain. This spectrum aids in identifying the frequency band where the sinusoidal noise resides. The Butterworth Band Reject Filter is chosen as the appropriate filter, designed to suppress these frequencies while preserving the rest of the image. The filter is crafted by defining a radial distance-based mask with inner and outer cutoff frequencies (`low_sigma` and `high_sigma`), which were determined through trial and error by visually inspecting the magnitude spectrum to match the noise band thickness. Frequencies within this band are zeroed out, leaving other frequencies unaltered.

The filtered frequency domain data is transformed back into the spatial domain using the inverse Fourier Transform to reconstruct the denoised image. The effectiveness of the Butterworth filter is evident from the reduced noise in the processed image, as well as the clear suppression of specific frequency bands in the updated magnitude spectrum. Parameters like the filter order and sigma values were iteratively adjusted to balance noise removal with image detail retention. The denoised image demonstrates how frequency-domain techniques can effectively remove structured noise without significantly degrading image quality.

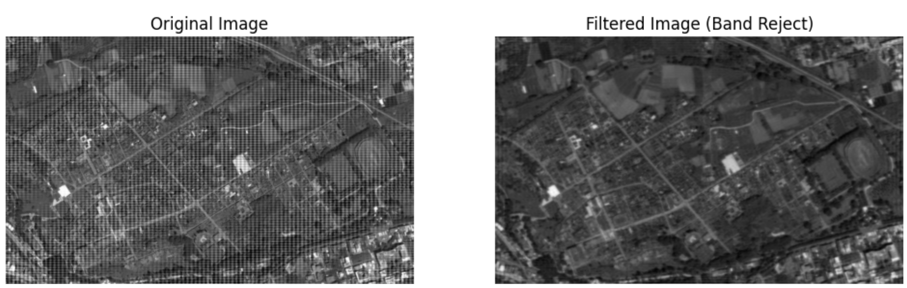

Figure 2: The Butterworth band reject filter application

3 Using Granulometry to Distinguish Gravels

This task involves matching unlabeled images to labeled ones based on granulometric histograms. Images undergo preprocessing, morphological operations, and histogram normalization. Euclidean distance compares feature vectors for matching. The granulometric histograms have been used to match the images of gravels with different size distributions. The histograms are the result of both standard openings and closings applied images separately.

(a) Are openings sufficient given brightness variability?
No, openings alone are insufficient when objects can be brighter or darker than their surroundings. Openings remove bright objects smaller than the structuring element, but they fail to address darker regions. To handle variability in brightness, **closings must be used in conjunction with openings**. This combination ensures that both bright and dark features are accounted for. Moreover, the brightness normalization through **histogram equalization** performed in preprocessing further mitigates the effect of brightness variability, making subsequent morphological operations more effective.

(b) What shape and size of structuring elements should be used?
Shape: Circular structuring elements are ideal because they align with the isotropic nature of many natural features in images.

Size/Range: Structuring elements should have radii that range from small to large, incrementing progressively until the morphological operations (opening and closing) no longer alter the image independently.

This approach ensures that features of all sizes in the image are captured. The iterative application of these operations with increasing radii effectively captures the full spectrum of feature sizes, with granulometric histograms reflecting the size distribution of objects in the image.

(c) Should openings/closings or openings/closings by reconstruction be used?
For this implementation, standard openings and closings were used to construct granulometric histograms due to their computational efficiency and sufficiency for capturing size distributions in the images. While openings and closings by reconstruction offer superior shape and connectivity preservation, they were not utilized as the primary objective was size-based histogram matching rather than detailed shape analysis. Standard operations were deemed adequate for this context.

(d) How it was supposed to compare the resulting numerical series against the granulometry of a labeled image?

Euclidean distance is used to compare granulometric histograms because it quantifies the magnitude of differences across entire histograms, offering an intuitive similarity measure. Other methods, like Manhattan distance, sum absolute differences, but may miss finer structural nuances. Chebyshev distance focuses on maximum difference in any bin, potentially overlooking overall distribution. The Euclidean distance is preferred for its balanced approach, capturing both overall and detailed differences in the histograms.

(e) Discuss how you could solve this problem using the frequency domain for this problem

The frequency domain offers an alternative way to analyze and compare images:

Fourier Transform: Can highlight repetitive patterns (e.g., texture) and filter out noise or unwanted frequencies.

Band-Pass Filters: Allow focusing on specific size ranges corresponding to the objects of interest.

Power Spectrum Analysis: Provides a quantitative representation of the image's texture and structure, which can be compared directly.

Hybrid Approach: Combine morphological analysis with frequency-based filtering to emphasize relevant patterns while suppressing irrelevant ones. Using the frequency domain, particularly with band-pass filtering or power spectrum analysis, could reduce computational complexity for texture-based comparisons and complement morphology-based techniques.

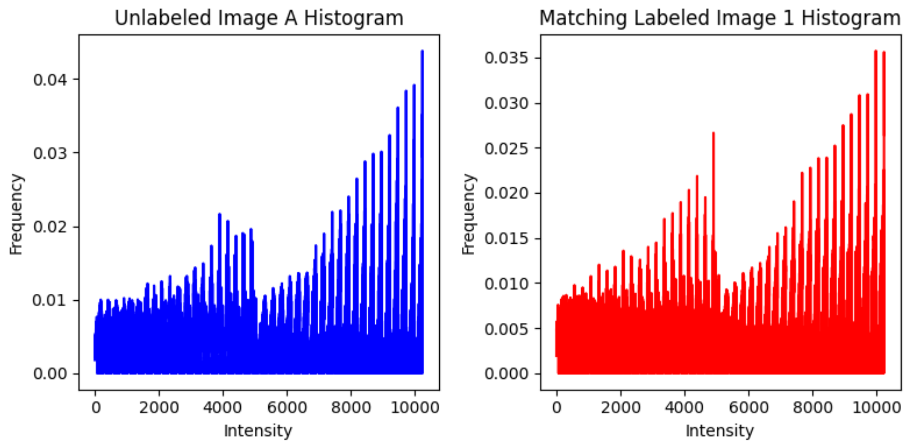

Figure 3: Example Granulometric Histogram Matching

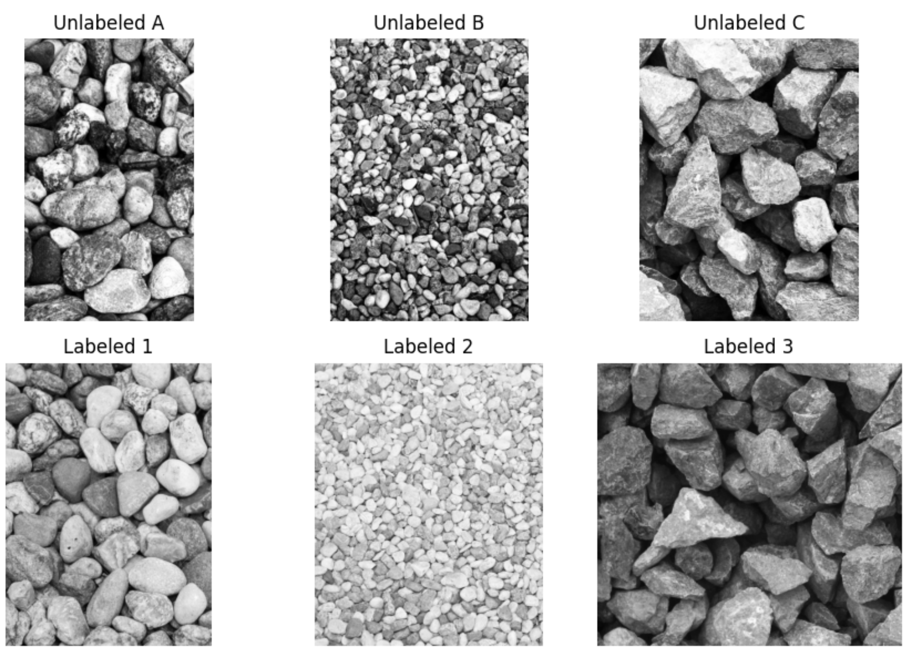

Figure 4: Matched Images

4 Median Filtering to Clear the Salt and Pepper Noise

In this section, the colored images were uploaded, and the test images were verified as 24-bit by checking the number of channels (for RBT, 3). Then, for each image, the salt and pepper effect was applied to corrupt it, and these test images were appended to the noisy_images array by selecting the different salt and pepper probabilities given below.

	Image	**Salt Probability**	**Pepper Probability**
0	test_image_1.jpg	0.02	0.07
1	test_image_2.jpg	0.10	0.02
2	test_image_3.jpg	0.07	0.08
3	test_image_4.jpg	0.10	0.07
4	test_image_5.jpg	0.02	0.02
5	test_image_6.jpg	0.03	0.09
6	test_image_7.jpg	0.09	0.03
7	test_image_8.jpg	0.07	0.05
8	test_image_9.jpg	0.02	0.05
9	test_image_10.jpg	0.06	0.03

Table 1: Salt and Pepper Probabilities for Each Image

Furthermore, the marginal and vector strategies on median filtering were implemented and applied to the noisy images.

4.1 Marginal Strategy Median Filter

The marginal_median_filter function applies a median filter independently to each channel of a multi-channel image (e.g., RGB). It pads the image for border handling, iterates over each pixel in every channel, extracts a local window defined by window_size, computes the median of the window, and updates the pixel value. This method processes each channel separately, preserving individual channel properties but ignoring relationships between channels, potentially leading to inter-channel inconsistencies. Note that the window is used to mention "kernel" here. Additionally, the results seem to be suitable for the denoising of the corrupted images.

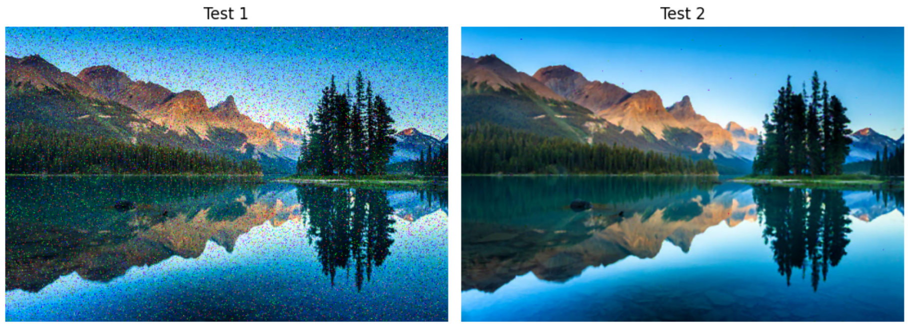

Figure 5: Marginal Median Filtered Image

4.2 Vector Strategy Median Filter

In this strategy, there are 3 orderings implemented as explained below.

4.2.1 Lexicographical Ordering

The lexicographical_vector_strategy function filters an image by replacing each pixel with the median pixel from its local neighborhood, determined using lexicographical ordering. Each pixel is treated as a vector (e.g., [R, G, B]), and its neighborhood is extracted, sorted lexicographically, and the middle value is selected. Padding ensures edge cases are handled, and the method processes multi-channel data cohesively. However, the result of the operation is not the expected since the noises seem more distributed and bigger but not denoised.

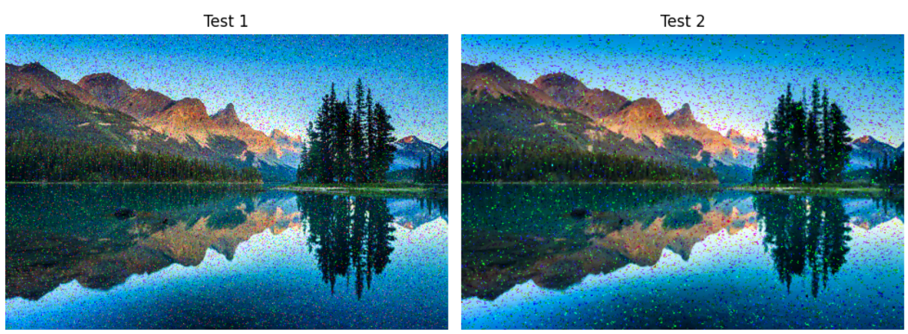

Figure 6: Lexicographical Ordering Median Filtered Image

4.2.2 L1 Norm Based Ordering (Manhattan Distance)

The l1_vector_strategy function applies a filtering technique to an image using the L1 (Manhattan) distance. It processes the image by extracting local windows around each pixel, where the window size must be odd. The image is padded with reflection mode to handle edges. For each pixel, the function computes the L1 distance between the reference pixel (the center of the window) and all other pixels in the window, sorting them based on these distances. The median pixel in the sorted list is then assigned to the corresponding pixel in the filtered image. Moreover, the result of the filtered image has changed but negatively, since the noise has not been cleaned perfectly.

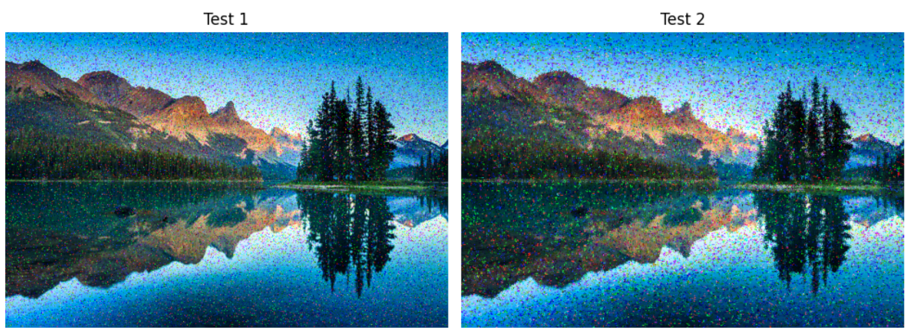

Figure 7: L1 Norm Based Ordering Filtered Image

4.2.3 L2 Norm Based Ordering (Euclidean Distance)

The l2_vector_strategy function applies a filtering technique to an image using the L2 (Euclidean) distance. Similar to the previous strategy, it processes the image by extracting local windows of pixels around each target pixel, where the window size must be odd. The image is padded using reflection mode to handle edge cases. For each pixel, the function computes the L2 distance between the reference pixel (the first pixel in the window) and all other pixels within the window. The pixels are then sorted based on these distances, and the median pixel from the sorted list is assigned to the filtered image. The pixel values are clipped to ensure they remain within the valid range (0-255). Even though the noise in the images seems to be decreased, it is not denoised well.

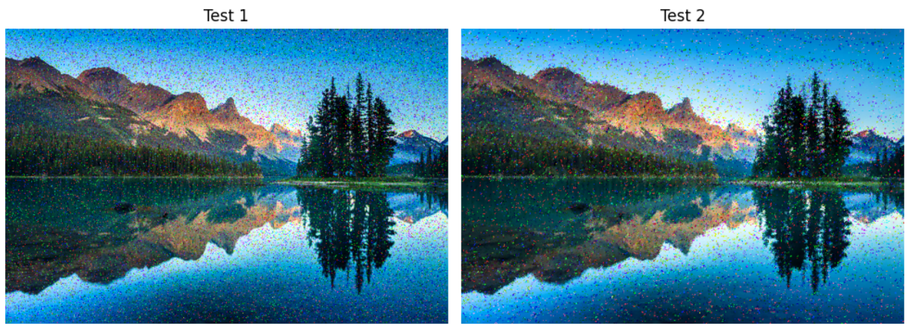

Figure 8: L2 Norm Based Ordering Filtered Image

	Image	Marginal MSE	Lexicographical MSE	L1 MSE	L2 MSE
0	test_image_1.jpg	22.904838	29.964175	49.712094	33.137814
1	test_image_2.jpg	8.081418	14.997538	36.134742	18.477567
2	test_image_3.jpg	37.783820	44.174843	65.077311	49.449193
3	test_image_4.jpg	10.771255	19.677871	40.255933	25.494407
4	test_image_5.jpg	22.651760	29.500438	54.624361	40.234991
5	test_image_6.jpg	9.424065	18.823534	40.363669	20.871661
6	test_image_7.jpg	14.212860	24.161191	43.008616	27.751197
7	test_image_8.jpg	50.577234	56.954069	79.626127	62.403866
8	test_image_9.jpg	30.376054	35.563555	60.030900	40.510056
9	test_image_10.jpg	12.396426	18.868311	32.154868	21.735139

Table 2: MSE Values for Each Image

a) Which one filters the image best, and by how much?

The best filter seems to be the Marginal Median Filter in terms of denoising the corrupted images visually when observed from the test images. Hence, according to the table above, MSE (Mean Squared Error) values between the original images (without any salt and pepper effect) and the filtered images, the least MSE values belong to the Marginal MSE. Thus, this supports the visual difference of denoising the images. Additionally, the data has been visualized and shown below for an easy comparison.

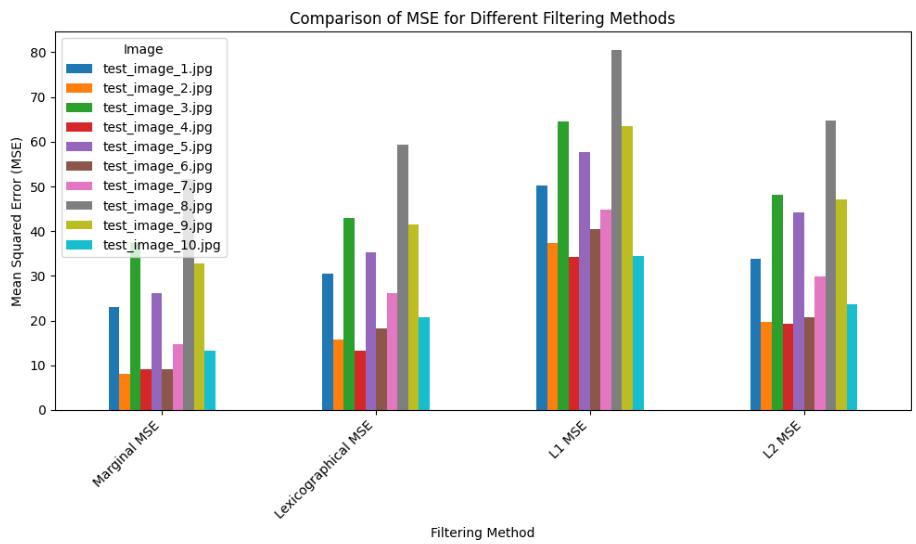

Figure 9: MSE Values Compared

b) Does their relative performance depend on the image?
As we can observe from Figure 9 and Table 2, we have different values for different images and different filters. Since the salt and pepper effect has been applied to the test images in various ratios and, they have different color distributions, it is normal to have different MSE values. However, relatively they have similar changes in MSE when the filters changed since we have the same bar chart structure for test images in different filters.

c) Or on the level/correlation of noise?
Having said that the salt and pepper effect has impacts on the performance of filters, it has been examined with Pearson Correlation Coefficient Calculation. The table formed below:

Correlation Pair	**Correlation Value**
Salt and Marginal MSE	0.4012259908386374
Pepper and Marginal MSE	0.3630519896375672
Salt and Lexicographical MSE	0.4213846610128144
Pepper and Lexicographical MSE	0.4519482573472859
Salt and L1 MSE	0.4273393701443463
Pepper and L1 MSE	0.37140858095872964
Salt and L2 MSE	0.3915613770401191
Pepper and L2 MSE	0.42757338419326907

Table 3: Correlation between Salt/Pepper and MSE for Different Filters

The correlation analysis reveals that both salt and pepper noise have a positive relationship with the Mean Squared Error (MSE) values across different filter types. As the amount of salt and pepper noise increases, the filter performance deteriorates, resulting in higher MSE values. Salt noise has a moderate to strong correlation with all MSE types, especially in Lexicographical and L1 filters, while pepper noise shows a slightly stronger effect on Lexicographical MSE. These findings suggest that both types of noise negatively impact filter performance, with pepper noise having a more significant influence on some filters like Lexicographical, while salt noise tends to affect others more evenly.

d) What about the effect of the filter size?

After testing the marginal median filter on test image 0, it was observed that applying the filter with a window size of 5 visually cleaned the image corruption better compared to using a window size of 3. However, the MSE results indicated that the smaller window size had a more favorable impact on preserving the original image. Specifically, for test image 0, the marginal median filter with a window size of 3 resulted in an MSE of 23.10, while the same filter applied with a window size of 5 produced an MSE of 30.58, highlighting the negative effect of the larger window size on image preservation.

e) What about the effect of the color space?

The test images were filtered base on the RBT color space. However we can examine the effect of the color spaces based on knowledge. The performance of median filtering can vary significantly depending on the color space used. In RBT, which separates brightness and chromatic components, median filtering might more effectively reduce noise in brightness without overly affecting color details. In CMYK, designed for print, median filtering may distort colors due to the interaction between the color channels. The CIELAB space, being perceptually uniform, may allow for more natural results with better noise reduction. Meanwhile, in color spaces like HSV or HLS, which separate hue and saturation from value, median filtering can preserve color information more effectively.