CS419 Digital Image and Video Analysis
Fall 2024
Assignment 3

Melih Kaan Şahinbaş

27.11.2024

Contents

1 Question 1

In Question 1, one of the hole-filling algorithms covered in the mathematical morphology topic is used. To fill the holes in the tire, the **Reconstruction by Erosion** technique is employed. This technique applies opening until $f(t) = f(t + 1)$ is achieved (f denotes Reconstruction by Erosion).

First, a marker is used, and then all values except the edges are assigned to the maximum value of the image's pixels. After that, the reconstruction by erosion function is called. Its result and the test image are printed to the console. In the reconstruction by erosion function, OpenCV's erosion function is utilized. As seen in the provided notebooks, OpenCV's implementation and custom implementation produce similar output. Therefore, the custom erosion function provided in the OpenCV documentation is not used. Below is a screenshot of the result.

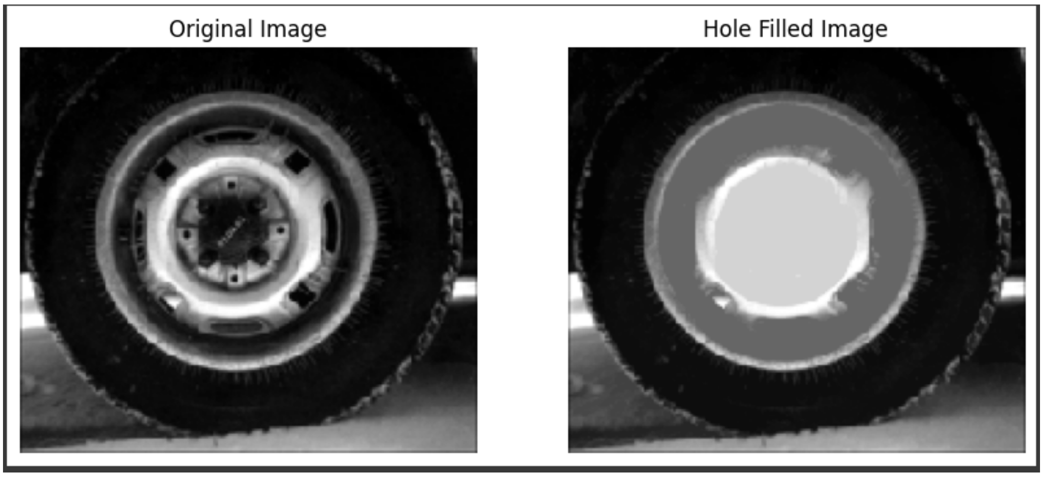
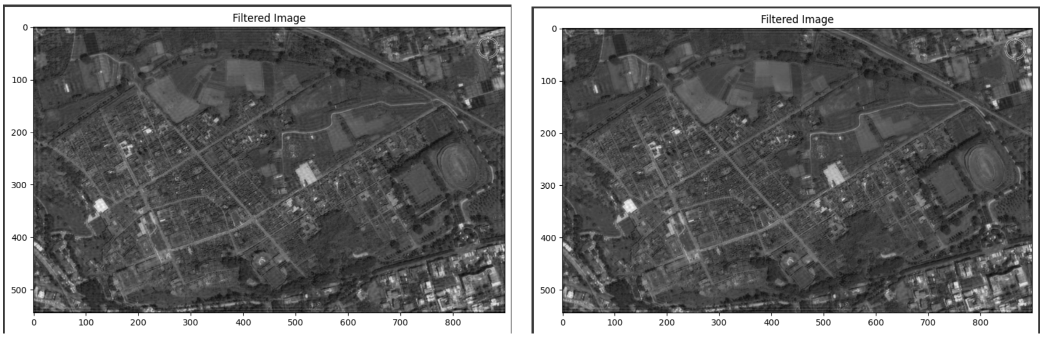


Figure 1: Result of Reconstruction by Erosion

2 Question 2

In Question 2, periodic noise is addressed using frequency techniques. After converting the image to the frequency domain using the fast Fourier transform and shifting, an **Ideal Band Reject Filter** is applied to the image with $D = 100$ and $W = 20$ (best result achieved with these parameters). An inverse shift and inverse fast Fourier transform are then applied, as shown in the Week-5 notebook.

I also tried the **Butterworth Band Reject Filter**. As seen in the lecture slides, this filter produces significantly successful results. Both methods resulted in similar mean squared error (MSE) compared to the base image, but the Butterworth Band Reject Filter had a slightly better outcome in terms of MSE. Both methods effectively handled the periodic noise. Below is a screenshot of the results.



(a) İdeal Band Reject Filtering Result (b) İdeal Band Reject Filtering Result

Figure 2: Results of frequency based filtering

3 Question 3

3.1 The objects of interest are sometimes brighter and sometimes darker with respect to their surroundings. Is using opening going to be sufficient?

Only using opening might not be sufficient when the objects of interest are brighter or darker with respect to their surroundings. As stated in the question above, brightness level might differ in each image at the different parts. To solve this problem **the Top-Hat Operator** might be used as shown in the lecture. The Top-Hat Operator operator highlights the significance of peaks, reducing the impact of background. Bright features are more easily identified with Top-Hat Operator. In this question granulometry found labels of unlabeled images correctly without using Top-Hat Operator. However, some photos might not be sufficiently labeled using opening instead of Top-Hat Operator.

3.2 What shape/range of sizes of structuring elements should you use?

The sizes and shapes of the particles and characteristics from which we want to extract information determine the size and shape of the structuring elements (SE) used in morphological procedures. In our problem, there are 3 types of different gravels with different sizes. Selection of the correct size of sieve is one of the most crucial features of granulometry. It allows robust shifting operation. First and last images have larger gravel sizes compared to the second image. However, there is a slight difference in their sizes. Therefore, applying different sizes of sieves from smaller values to bigger values might be efficient to correctly label the unlabeled images. From 1 to 15 different radius' are used.

In terms of shape, the gravels are eliptic or circular shaped so using circular structuring element with different radius' might give the correct result. At the end, correct classification is achieved.

3.3 Should you use openings/closings or openings/closings by reconstruction?

It is possible to implement granulometries with the above operations. Moreover, openings/closings by reconstruction provides better implementation compared to openings/closings. As we covered in CS419, opening by reconstruction can further smoothen the image with a certain extra computational cost. Since this question does not require such an attempt, I used opening instead of opening by reconstruction. Consequently, accurate classification results are obtained.

Last but not least, opening by reconstruction can be used if needed. However, if the task can be handled without opening by reconstruction, it would be better not to use it due to extra computational cost of it.

3.4 As if that wasn't enough, say you calculate the granulometry of an unlabeled image, how are you supposed to compare the resulting numerical series against the granulometry of a labeled image? Should you rely on the Euclidean distance ? Or their Manhattan distance? Or Chebyshev distance? Or something entirely different?

For each unlabeled image, the algorithm calculates the difference between the granulometry of each labeled picture and the granulometry of each unlabeled image. Three metrics—**Euclidean Distance**, **Manhattan Distance**, and **Chebyshev Distance**. —are used to determine the distances.

For each unlabeled image, the algorithm finds the identified image that is closest to the unlabeled image. According to that measure, this labeled image is the closest match.

The closest labeled image for each distance metric is reported, associating the unlabeled image with the most similarly labeled image based on its granulometry. This approach offers a methodical manner for classifying the unlabeled images by guaranteeing that every unlabeled image is thoroughly evaluated against every labeled image.

For this sample set, all of the three metrics—**Euclidean Distance**, **Manhattan Distance**, and **Chebyshev Distance** were able to classify images correctly.

As shown below:

Unlabeled Image	**Euclidean Distance**	**Manhattan Distance**	**Chebyshev Distance**
/unlabeled-A.jpg	Labeled-1	Labeled-1	Labeled-1
/unlabeled-B.jpg	Labeled-2	Labeled-2	Labeled-2
/unlabeled-C.jpg	Labeled-3	Labeled-3	Labeled-3

Table 1: Error Results for Unlabeled Images and Closest Labeled Matches

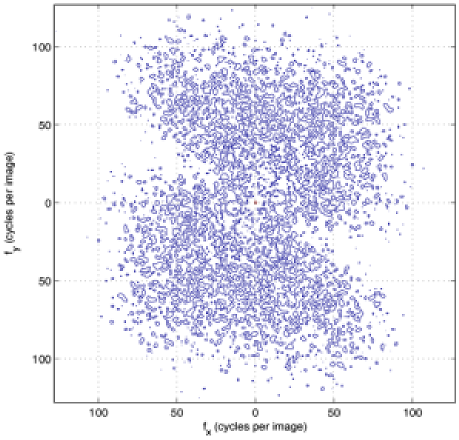
3.5 Discuss how you could solve this problem using the image's frequency domain representation (with or without morphology)?

Each picture can be transformed into the frequency domain using the 2D Fourier Transform, allowing for an analysis of the power or energy distribution throughout the low, middle, and high frequency bands. With coarse textures predominating at low frequencies and delicate textures displaying greater high-frequency components, these frequency bands capture distinctive texture properties. The closest labeled match for each unlabeled picture can be determined by computing and comparing the energy in these bands for each labeled and unlabeled image using distance measures like the Euclidean distance. This method can work well for categorizing gravel according to its granulometry since it is noise-resistant and efficient at detecting recurring or periodic patterns in the textures.

"Since frequency is related directly to rate of change, frequencies in the Fourier transform can be associated with the patterns of intensity variations in an image." and "High frequencies correspond to fastly varying components of the image such as edges and noise." These are from lecture slides.

These are also from lecture slides, using frequency in both directions, gravel sizes can be determined. For example, smaller gravels will have more high frequencies than the bigger gravels. And above method can be implemented using this idea.

(a) From lecture 1



(b) From lecture 2

Figure 3: Two images side by side

4 Question 4

In question 4, I applied salt and pepper noise adding and median filtering with 4 different ordering techniques such as **Marginal**, **Lexicographical**, **L1 Norm**, and **L2 Norm** ordering techniques. These techniques were used on 10 different images found on the Internet. First salt and pepper noise is added to these images and after that median filtering with different ordering techniques (aforementioned methods) are used to these noised images in order to reduce noise. These techniques are tested with different noise, kernel size and color space in order to examine the effects of these parameters to median filtering. These results are obtained in the python notebook and added to this report.

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

According to the mean square error (MSE) results of the tests, the marginal median filter removed the noise better than the other filters used as seen below (put results as a table). I applied these filters to 24-bit RGB images with kernel size 3 and noise 0.1.

As it can be seen from table 1, the marginal median filter performs better than others in each image with smaller MSE values. The numerical results indicated above how much the marginal median filter performs better than other filters.

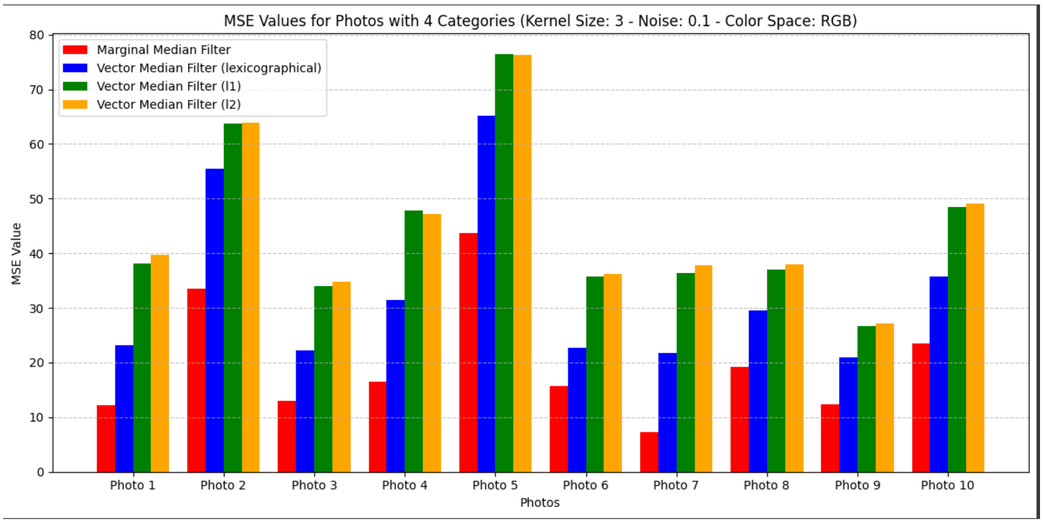


Figure 4: Performance Comparison of Filtering Techniques

4.1 Does their relative performance depend on the image?

To answer this question, the mean square error results from previous subquestion might be considered. As seen in the table, marginal median filtering and lexicographical median filtering has uniform ranking. Marginal median filtering always outperforms others and lexicographical median filtering always outperforms L1 norm based and L2 norm based techniques. However, from image to image L1 and L2 norm based techniques have different rankings. In some images L1 has better mean square errors and in some images L2 has better mean square errors.

4.2 Or on the level/correlation of noise?

To answer this question, I applied the same filters with 0.3 noise. The results show that when noise increases image quality decreases and also we observe a certain increase in Mean Square Error results. It is expected because, when noise increases, the impact of the filter decreases relative to smaller noises. Given that the median filter works by extracting the median value from a sorted list, it is quite likely to become confused and place deteriorating pixels in the center of the kernel if there is a lot of noise. Below you can see the total amount of MSE's per filtering technique compared to the smaller noise size.

Table 2: Performance Comparison of Filtering Techniques at Different Noise Levels

Noise Level	Marginal	Lexicographical	L1 Norm	L2 Norm
Noise = 0.1	197.2097	328.3099	444.3149	450.1226
Noise = 0.3	246.1178	338.7508	544.1703	556.8864

4.3 What about the effect of the filter size?

To answer this question, I applied the same filters with kernel size 5. In return, I obtained that smaller kernel sizes perform better than larger kernel sizes in terms of MSE. Smaller kernels are less influenced by their surrounding noise compared to bigger kernels. Therefore, this causes more error in the filtered image compared to the smaller kernel size and results in higher MSE results for images. Below you can see the results: .

Table 3: Performance Comparison of Filtering Techniques at Different Kernel Sizes

Kernel Size	Marginal	Lexicographical	L1 Norm	L2 Norm
Kernel Size = 3	197.2097	328.3099	444.3149	450.1226
Kernel Size = 5	290.3133	377.6604	475.6321	479.0985

4.4 What about the effect of the color space?

As shown in the results L*A*B color space has significantly better MSE error results in every filtering technique. However, since every color space has its own properties other color spaces' MSE results migt be different.

Table 4: Performance Comparison of Filtering Techniques in Different Color Spaces

Color Space	Marginal	Lexicographical	L1 Norm	L2 Norm
RGB	197.2097	328.3099	444.3149	450.1226
LAB	79.4659	152.9037	240.0339	232.0380