**Digital Image Processing Laboratory**

Experiment Report

Experiment Title Smoothing filter and sharpening filter

Student’s Name 蒋云翔

Student’s ID 2022102330

Class CST

Date handed in 2024/11/13

International school, Jinan University

**A. Objectives**

1. To know how to smooth or sharpen images
2. To be able to implement simple smooth or sharpening filter in MATLAB

**B. Technique**

In this lab, the image **lib.jpg** will be used.

1. Add noise to the image.
2. Apply average filter, Gaussian filter and custom filter to smooth the image.
3. Apply Laplacian mask to get the Laplacian image, then perform image addition.
4. Use Robert operator and Sobel operator to get the gradient image.
5. Design one sharpening mask to sharpen the image

**C. Experiment Content**

1. **Smoothing image**
2. Read the images **lib.jpg** .

建筑的摆设布局

中度可信度描述已自动生成

Figure 1: The original Image

1. Convert it into grey image.



Figure 2: The original gray Image

1. Add some noise on the grey image (name it as **GIMGNoise**). (Using **imnoise** function)

建筑与房屋的城市空拍图

描述已自动生成

Figure 3: The GIMG with noise (GIMG)

1. Using averaging filter to smooth the image **GIMGNoise**. Use different size of averaging mask to do it and compare the results.



建筑的摆设布局

描述已自动生成

Figure 4: Using averaging filter with different size to smooth the image GIMGNoise

The above figures show the effect of smoothing an image using averaging filters of different sizes. From top to bottom, as the mask size increases (3x3 to 5x5,), it can be observed that the noise of the image decreases and the details become blurrier. This suggests that a larger mask size will smooth the image more effectively, but at the expense of image sharpness and detail.

Therefore, in practical applications, choosing the appropriate mask size to achieve the best balance between denoising and preserving details is a key consideration in image processing

1. Using Gaussian filter to smooth the image **GIMGNoise**. Use different size of mask to do it. After that, compare the results from Gaussian filter.

建筑的截图

描述已自动生成

Figure 5: Using Gaussian filter with different Sigma and same filter size to smooth the image GIMGNoise

These above figures show the results of smoothing an image using a Gaussian filter, using different σ (standard deviation) values but keeping the same mask size (3x3). As the σ value increases from 0.5 to 1.0, the smoothing effect of the image gradually increases and the noise decreases.

However, as the σ value increases, the details of the image also decrease accordingly. This shows that the σ value of the Gaussian filter has a significant impact on image smoothness and detail retention

1. Design one smoothing mask to smooth the image **GIMGNoise** and show its results.

建筑的摆设布局

描述已自动生成

Figure 6: Using custom smoothing mask to smooth the image GIMGNoise

The smoothing mask I designed uses a weighted average method, with the center pixel having the greatest weight and decreasing weight as the pixels get further away from the center.

This design aims to **take more into account the influence of the central area of the image**, while also **slightly considering surrounding pixels**, thereby smoothing out noise while maintaining some sharpness in the image.

According to the results shown, the mask can **effectively reduce noise and the image looks smoother**, but still retains some details. The advantage of this mask is that it provides a balanced method between a simple average filter and a Gaussian

filter, which has good effects on detail preservation and noise removal in the image

The code to implement the above functions is as follows:

|  |
| --- |
| % 1. Read the image  img = imread("C:\Users\86136\Desktop\Digital Image processing\Lab\DIP LAB\lab 3\lib.jpg");  % 2. Display the original color image  figure;  subplot(3, 2, 1);  imshow(img);  title('Original Color Image');  % 3. Convert to grayscale image  gray\_img = rgb2gray(img);  % Display the original grayscale image and the image with noise  subplot(3, 2, 2);  imshow(gray\_img);  title('Original Gray Image');  % 4. Add noise  GIMGNoise = imnoise(gray\_img, 'salt & pepper', 0.02);  subplot(3, 2, 3);  imshow(GIMGNoise);  title('Image with Noise');  % 5. Smooth the image using the averaging filter (3x3 and 5x5 masks)  avg\_filter\_3x3 = fspecial('average', [3 3]);  avg\_filtered\_img\_3x3 = imfilter(GIMGNoise, avg\_filter\_3x3);  avg\_filter\_5x5 = fspecial('average', [5 5]);  avg\_filtered\_img\_5x5 = imfilter(GIMGNoise, avg\_filter\_5x5);  subplot(3, 2, 4);  imshow(avg\_filtered\_img\_3x3);  title('Averaging Filter 3x3');  subplot(3, 2, 5);  imshow(avg\_filtered\_img\_5x5);  title('Averaging Filter 5x5');  % 6. Smooth the image using the Gaussian filter with varying sigma values  figure;  % Gaussian filter with 3x3 mask and sigma = 0.5  gaussian\_filter\_3x3\_sigma\_0\_5 = fspecial('gaussian', [3 3], 0.5);  gaussian\_filtered\_img\_3x3\_sigma\_0\_5 = imfilter(GIMGNoise, gaussian\_filter\_3x3\_sigma\_0\_5);  subplot(3, 2, 1);  imshow(gaussian\_filtered\_img\_3x3\_sigma\_0\_5);  title('Gaussian Filter 3x3, Sigma 0.5');  % Gaussian filter with 3x3 mask and sigma = 1  gaussian\_filter\_3x3\_sigma\_1 = fspecial('gaussian', [3 3], 1);  gaussian\_filtered\_img\_3x3\_sigma\_1 = imfilter(GIMGNoise, gaussian\_filter\_3x3\_sigma\_1);  subplot(3, 2, 2);  imshow(gaussian\_filtered\_img\_3x3\_sigma\_1);  title('Gaussian Filter 3x3, Sigma 1');  % Gaussian filter with 5x5 mask and sigma = 0.5  gaussian\_filter\_5x5\_sigma\_0\_5 = fspecial('gaussian', [5 5], 0.5);  gaussian\_filtered\_img\_5x5\_sigma\_0\_5 = imfilter(GIMGNoise, gaussian\_filter\_5x5\_sigma\_0\_5);  subplot(3, 2, 3);  imshow(gaussian\_filtered\_img\_5x5\_sigma\_0\_5);  title('Gaussian Filter 5x5, Sigma 0.5');  % Gaussian filter with 5x5 mask and sigma = 1  gaussian\_filter\_5x5\_sigma\_1 = fspecial('gaussian', [5 5], 1);  gaussian\_filtered\_img\_5x5\_sigma\_1 = imfilter(GIMGNoise, gaussian\_filter\_5x5\_sigma\_1);  subplot(3, 2, 4);  imshow(gaussian\_filtered\_img\_5x5\_sigma\_1);  title('Gaussian Filter 5x5, Sigma 1');  % 7. Design a custom smoothing mask  custom\_mask = [1 2 1; 2 4 2; 1 2 1] / 16;  custom\_filtered\_img = imfilter(GIMGNoise, custom\_mask);  % Display the image smoothed with the custom mask  subplot(3, 2, 5);  imshow(custom\_filtered\_img);  title('Custom Smoothing Mask');  % Display the original noise image  subplot(3, 2, 6);  imshow(GIMGNoise);  title('Original Noise Image'); |

1. **Sharpening Images**
2. Read the images **lib.jpg** given in the folder.

建筑的摆设布局

中度可信度描述已自动生成

Figure 7: The original Image

1. Convert it into grey image **GIMG**.



Figure 8: The original gray Image

1. Use Laplacian mask to get the Laplacian image of **GIMG**. Laplacian mask has four types. Try to use each one to do it.

The following are the contents of the four Laplacian masks:

黑色的钟表

中度可信度描述已自动生成图示, 示意图

描述已自动生成

Mask 1 Mask 2

钟表的特写

中度可信度描述已自动生成文本

低可信度描述已自动生成

Mask 3 Mask 4

图示

中度可信度描述已自动生成

Figure 9: Use Laplacian mask 1 to get the Laplacian image of GIMG

This is a common 4 neighborhood Laplacian mask. This mask emphasizes the difference between the center pixel and its four neighboring pixels, which are right up, right down, right left and right, and is used to enhance the edge features of the image. Since only pixels in four directions are considered, noise amplification is less than other masks

电子设备的屏幕

中度可信度描述已自动生成

Figure 10: Use Laplacian mask 2 to get the Laplacian image of GIMG

This is a more comprehensive 8-neighborhood Laplacian mask that considers all eight orientations around the center pixel. This mask enhances edge detail more significantly, but also amplifies noise more. This comprehensive mask is very effective for detecting subtle edges.

图形用户界面

描述已自动生成

Figure 11: Use Laplacian mask 3 to get the Laplacian image of GIMG

This is a variant of the Laplacian mask where the center pixel has a weight of 4 and the surrounding pixels have a weight of ±2. This mask is designed to detect edges and turning points more aggressively, and is better at eliminating certain types of noise, but also amplifies stronger edges.



Figure 12: Use Laplacian mask 4 to get the Laplacian image of GIMG

This is a Laplacian mask of weighted centers, with the center pixels having weights of 12 and the surrounding pixels having weights of ±1 and ±2. This mask puts more emphasis on the difference between the center pixel and the surrounding pixel, can be used to detect stronger edge changes, and more significantly enhance the high-frequency components in the image, suitable for scenes that require significant edge enhancement.

1. Add the four Laplacian images on the image **GIMG**. Compare the four resulting images.

河边的城市

描述已自动生成

Figure 13: Add the first Laplacian image on the image GIMG

Edge features in the image are enhanced when we add the Laplacian mask 1 to the original image. Because this mask emphasizes the differences between the central pixel and its four directional neighborhood pixels, the resulting image has sharper edges and more prominent details. This helps detect and enhance boundaries in the image but does not greatly amplify the noise.

建筑与房屋的城市空拍图黑白照

描述已自动生成

Figure 14: Add the second Laplacian image on the image GIMG

After adding the Laplacian mask 2 to the original image, the edge features of the image become more significant. This is because this mask considers all eight orienting pixels around the center pixel, enabling a more comprehensive detection and enhancement of edges. The resulting image shows stronger edge detail, but at the same time may amplify the noise, making the image look sharper but slightly grittier.

建筑的摆设布局

描述已自动生成

Figure 15: Add the third Laplacian image on the image GIMG

Using Laplacian mask 3 to add images, edge features become sharper and more pronounced. This mask is designed to detect and enhance edges more aggressively, especially more intense edge changes. While this method can significantly improve edge detail, it can also introduce more noise and make parts of the image look too sharp.

图示, 工程绘图

描述已自动生成

Figure 16: Add the fourth Laplacian image on the image GIMG

After adding the Laplacian mask 4 to the original image, the edge features of the image are very significant. This is because mask 4 greatly emphasizes the difference between the central pixel and its neighboring pixels. Results The high frequency component in the image is obviously enhanced, which is suitable for the scene where the edge is strongly enhanced. Note, however, that this also amplifies the noise significantly, making the image appear very sharp.

1. Use Robert operator and Sobel operator to get the gradient image of **GIMG**.

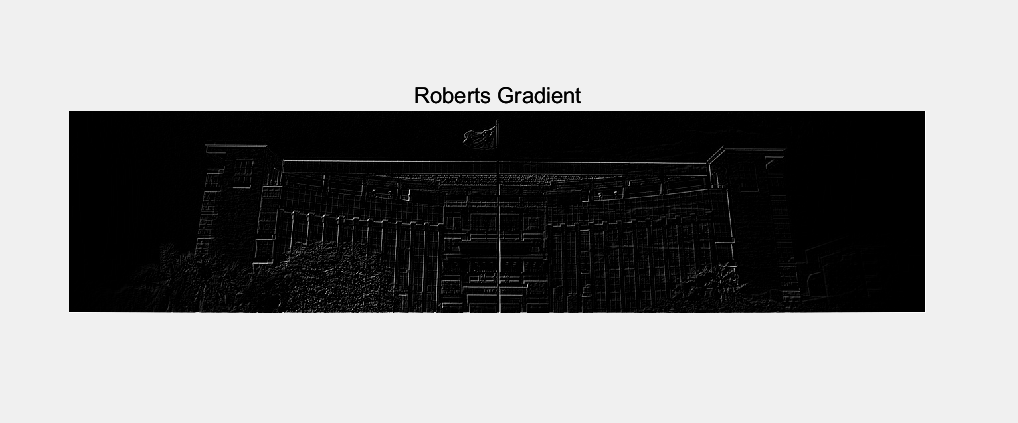


Figure 17: Use Robert operator to get the gradient image of GIM

Roberts operators are mainly used to detect the edges of images. It highlights the changing part of the image by calculating the difference between adjacent pixels, it can detect the edge in the diagonal direction (i.e. 45° and 135° directions) due to its simple structure, it is also sensitive to noise, will amplify the noise in the image, due to the small convolution kernel, less computation, suitable for fast processing applications.

电脑萤幕画面

中度可信度描述已自动生成

Figure 18: Use Sobel operator to get the gradient image of GIM

Sobel operator is also used for edge detection, but compared with Roberts operator, it can smooth and suppress noise better. It detects horizontal and vertical edges respectively and can combine the results of these two directions to generate the overall edge image. Due to the large convolution kernel, it smoothens the image more, thus reducing the influence of noise. The Sobel operator can better enhance the edges in the image and make the edges clearer

1. Design one sharpening mask to sharpen the image **GIMG** and show its results. Note: the mask depends on your idea, no any restriction.

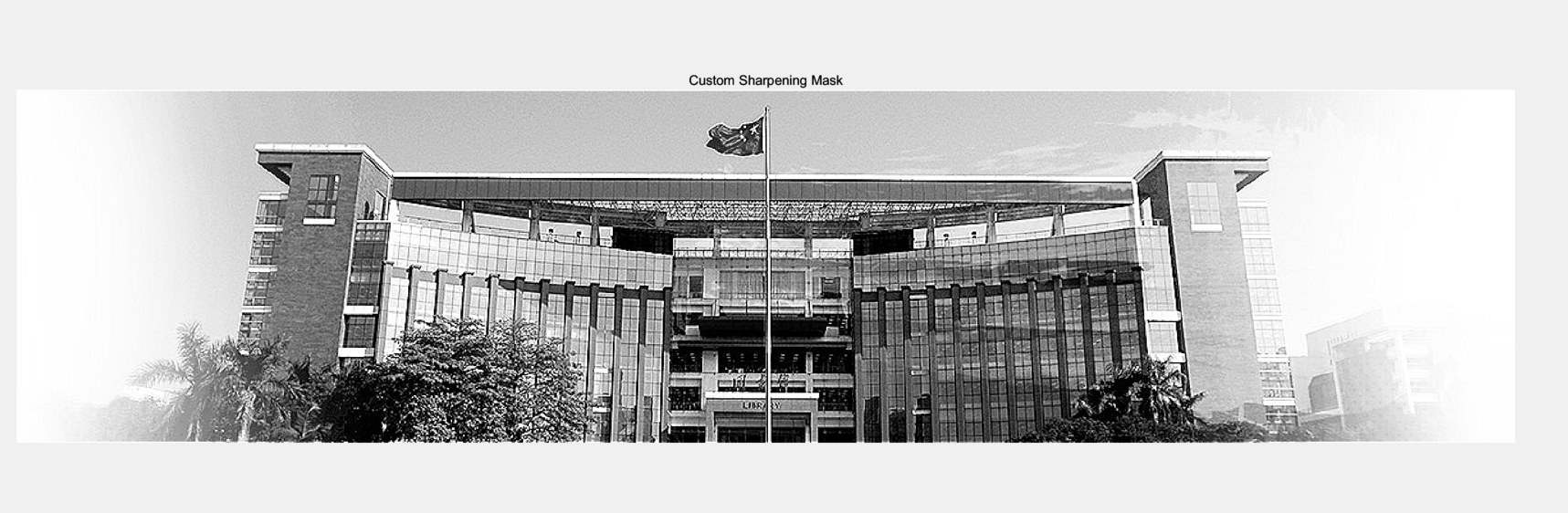


Figure 19: Using custom sharpening mask to sharpen the GIMG

This custom sharpening mask uses an 8-neighbor configuration with negative weights and a positive weight at the center. The structure of the mask is designed as follows:

黑色的钟表

描述已自动生成

Figure 20: Sharpening mask

**Functionality and Effect:**

1. **Brightness Enhancement**: The central pixel’s brightness is increased by the positive weight of 5, while the surrounding pixels' brightness is reduced by the negative weights of -1. This enhancement increases the local contrast around each pixel.
2. **Edge and Detail Enhancement**: By increasing the contrast between the central pixel and its neighboring pixels, the mask sharpens edges and makes details more prominent. This means that edges and textures in the image GIMG will appear sharper and more defined.
3. **Increased Contrast**: This mask improves the overall contrast of the image, especially in areas with significant color transitions. This can make objects and features in the image stand out more clearly.
4. **Noise Amplification**: One side effect of this sharpening process is that it also enhances high-frequency noise. This means that noise present in the original image can become more pronounced.
5. **Visual Artifacts**: Over-sharpening can sometimes lead to unnatural visual effects such as bright halos or dark rings around edges. This can result in the image looking slightly artificial if the sharpening effect is too strong.
6. **Enhanced Depth and Texture**: Visually, the sharpened image should exhibit more depth and texture compared to the original image. However, it is essential to balance the sharpening to avoid excessive artifacts, particularly in areas with complex details or high noise levels.

The code to implement the above functions is as follows:

|  |
| --- |
| % 1. Read the image  img = imread("C:\Users\86136\Desktop\Digital Image processing\Lab\DIP LAB\lab 3\lib.jpg");  % 2. Convert to grayscale image  GIMG = rgb2gray(img);  % Display the original grayscale image  figure;  subplot(3, 3, 1);  imshow(GIMG);  title('Original Gray Image');  % 3. Use Laplacian mask to get the Laplacian images of GIMG  laplacian\_masks = {  [0 -1 0; -1 4 -1; 0 -1 0], ... % Mask 1  [-1 -1 -1; -1 8 -1; -1 -1 -1], ... % Mask 2  [1 -2 1; -2 4 -2; 1 -2 1], ... % Mask 3  [-2 -1 -2; -1 12 -1; -2 -1 -2] % Mask 4  };  laplacian\_images = cell(1, 4);  for i = 1:4  laplacian\_images{i} = imfilter(GIMG, laplacian\_masks{i});  subplot(3, 3, i+1);  imshow(laplacian\_images{i}, []);  title(['Laplacian Mask ', num2str(i)]);  end  % 4. Add the Laplacian images on the image GIMG and compare the results  added\_images = cell(1, 4);  for i = 1:4  added\_images{i} = imadd(GIMG, laplacian\_images{i});  subplot(3, 3, i+5);  imshow(added\_images{i}, []);  title(['Added Laplacian Mask ', num2str(i)]);  end  % 5. Use Robert operator and Sobel operator to get the gradient image of GIMG  roberts\_operator = [1 0; 0 -1]; % Define Roberts operator  gradient\_roberts = imfilter(GIMG, roberts\_operator);  sobel\_operator = fspecial('sobel');  gradient\_sobel = imfilter(GIMG, sobel\_operator);  figure;  subplot(1, 2, 1);  imshow(gradient\_roberts, []);  title('Roberts Gradient');  subplot(1, 2, 2);  imshow(gradient\_sobel, []);  title('Sobel Gradient');  % 6. Design one sharpening mask to sharpen the image GIMG and show its results  custom\_sharpening\_mask = [0 -1 0; -1 5 -1; 0 -1 0];  sharpened\_image = imfilter(GIMG, custom\_sharpening\_mask);  figure;  imshow(sharpened\_image, []);  title('Custom Sharpening Mask'); |

**D. Conclusions**

In this series of digital image processing tasks, we have explored several fundamental techniques and filters to enhance, analyze, and improve the quality of images. Here’s a summary of what we accomplished and the key learnings from each task:

1. **Histogram Equalization**:
   * We performed histogram equalization on the image to enhance its contrast, making details more visible. By transforming the original image and comparing the histograms before and after equalization, we observed how this technique spreads out the most frequent intensity values, leading to a more balanced and visually appealing image.
2. **Noise Addition and Filtering**:
   * By adding 'salt & pepper' noise to a grayscale image, we simulated real-world noise interference. We then applied various filters, including averaging and Gaussian filters with different mask sizes, to smooth the noisy image. This task demonstrated the effectiveness of different filtering techniques in reducing noise while preserving important details.
3. **Laplacian and Gradient Masks**:
   * We used various Laplacian masks to detect edges in the grayscale image. Each mask emphasized different aspects of the image, showing how varying Laplacian masks can enhance or highlight different edge characteristics. Adding these Laplacian images to the original image allowed us to enhance edges and compare the effects of different masks.
   * We also applied Roberts and Sobel operators to detect gradients and highlight edges in the image. The Roberts operator provided simple edge detection, while the Sobel operator offered more comprehensive edge enhancement with better noise smoothing capabilities.
4. **Custom Smoothing and Sharpening**:
   * Designing a custom smoothing mask, we applied it to the noisy image to achieve specific smoothing effects based on our design. This task emphasized the flexibility and creative potential of designing custom filters tailored to specific needs.
   * We also created a custom sharpening mask to enhance the edges and details of the image. By applying this mask, we increased the local contrast and made textures more prominent, while being mindful of potential noise amplification and visual artifacts.

**Key Learnings**

* **Contrast Enhancement**: Histogram equalization effectively enhances image contrast, making details more visible across a wider intensity range.
* **Noise Reduction**: Averaging and Gaussian filters are powerful tools for reducing noise, each offering unique advantages in balancing noise reduction and detail preservation.
* **Edge Detection**: Different Laplacian masks and gradient operators highlight various edge characteristics, allowing for customized edge detection and enhancement.
* **Custom Filtering**: Designing and applying custom smoothing and sharpening masks provides flexibility in achieving desired image processing outcomes.

These tasks collectively provided a comprehensive overview of essential digital image processing techniques, equipping us with the tools and knowledge to tackle various image enhancement and analysis challenges in practical applications.