Addis Ababa University

Master's in Artificial Intelligence

Digital Image Processing (DIP) Laboratory Manual

Prepared by: Dr. Natnael Argaw Wondimu

Environment: GNU Octave

Preface

This manual is prepared in a workbook style for AI Master's students at Addis Ababa University. It provides a hands-on introduction to Digital Image Processing using GNU Octave, covering basic to advanced topics. Each lab includes objectives, theoretical background, procedures, Octave code, checkpoints, try-it-yourself prompts, and collaborative assignments. The manual is intended to build foundational knowledge in Digital Image Processing and serve as a base for advanced studies in Computer Vision.

Chapter 9: Noise Reduction and Restoration (GNU Octave)

Objective

To understand types of image noise and apply classical image restoration techniques such as filtering, Wiener filtering, and inverse filtering using GNU Octave.

1. Introduction to Image Noise

Description: Noise in digital images is an unwanted variation in pixel intensity due to acquisition errors, transmission, or sensor limitations. Common types include: - **Gaussian noise**: Random variations in brightness. - **Salt & Pepper noise**: Sparse white and black dots. - **Speckle noise**: Granular noise in radar/ultrasound images.

2. Adding Noise to an Image (For Simulation)

2.1 Gaussian Noise

Code Snippet:

```
pkg load image;
img = im2double(imread('cameraman.tif'));
gauss_img = imnoise(img, 'gaussian', 0, 0.01);
imshow(gauss_img);
title('Image with Gaussian Noise');
```

Output Description: Adds slight grainy brightness fluctuations across the image.

2.2 Salt & Pepper Noise

Code Snippet:

```
sp_img = imnoise(img, 'salt & pepper', 0.05);
imshow(sp_img);
title('Image with Salt & Pepper Noise');
```

Output Description: Random white and black pixels appear scattered throughout the image.

3. Noise Reduction Techniques

3.1 Mean (Averaging) Filter

Code Snippet:

```
kernel = ones(3,3)/9;
mean_filtered = imfilter(gauss_img, kernel);
imshow(mean_filtered);
title('Mean Filtered Image');
```

Output Description: Reduces overall noise but may blur fine details.

3.2 Median Filter (Best for Salt & Pepper)

Code Snippet:

```
median_filtered = medfilt2(sp_img);
imshow(median_filtered);
title('Median Filtered Image');
```

Output Description: Effectively removes salt and pepper noise while preserving edges.

3.3 Gaussian Filter

Code Snippet:

```
gaussian_filtered = imgaussfilt(gauss_img, 1);
imshow(gaussian_filtered);
title('Gaussian Filtered Image');
```

Output Description: Smooths the image and reduces high-frequency noise.

4. Image Restoration Techniques

4.1 Wiener Filter

Code Snippet:

```
wiener_img = wiener2(gauss_img, [5 5]);
imshow(wiener_img);
title('Wiener Filtered Image');
```

Output Description: Adaptive filtering that adjusts to local image variance. Performs well in Gaussian noise conditions.

4.2 Inverse Filtering (Deblurring)

Code Snippet:

```
PSF = fspecial('motion', 15, 45);
blurred = imfilter(img, PSF, 'conv', 'circular');
restored = deconvwnr(blurred, PSF);
imshow(restored);
title('Inverse Filtered (Deblurred) Image');
```

Output Description: Attempts to reverse blurring using the known point spread function (PSF).

5. Comparing Results

You can visually and quantitatively compare filter performance using metrics like MSE or PSNR.

Example:

```
mse = mean((img(:) - gaussian_filtered(:)).^2);
psnr = 10 * log10(1 / mse);
```

Output Description: Returns numerical quality measure of noise reduction performance.

6. Summary

- **Median filters** are ideal for impulse noise.
- **Gaussian and Mean filters** are suited for Gaussian noise.
- **Wiener filter** adapts based on local image statistics.
- **Inverse filtering** requires knowledge of degradation.

Suggested Exercises

1. Add noise of different types and levels; apply appropriate filters.
2. Compare PSNR values of all restoration results.
3. Implement motion blur followed by inverse filtering.
4. Combine spatial and frequency domain filtering for advanced restoration.