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 6: Image Transforms (GNU Octave)

Objective

To explore and implement common image transforms such as the Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), and Discrete Wavelet Transform (DWT) using GNU Octave.

1. What are Image Transforms?

Description: Image transforms convert an image from the spatial domain to the frequency or another domain. This is useful for analysis, compression, and filtering.

2. Preparing the Environment

2.1 Load Required Package and Image

Code Snippet:

```
pkg load image;
img = im2double(imread('cameraman.tif'));
imshow(img);
title('Original Image');
```

Output Description: Loads and displays a grayscale image for transformation.

3. Discrete Fourier Transform (DFT)

3.1 What is DFT?

Description: DFT converts an image from spatial to frequency domain, revealing frequency components.

3.2 Applying 2D Fourier Transform

Code Snippet:

```
F = fft2(img);
F_shifted = fftshift(F);
F_magnitude = log(1 + abs(F_shifted));
imshow(F_magnitude, []);
title('Magnitude Spectrum of DFT');
```

Output Description: Displays the frequency spectrum. Bright points indicate strong frequency components.

3.3 Inverse Fourier Transform

Code Snippet:

```
reconstructed = real(ifft2(F));
imshow(reconstructed);
title('Reconstructed Image from DFT');
```

Output Description: Recovers the original image from the frequency domain.

4. Discrete Cosine Transform (DCT)

4.1 What is DCT?

Description: DCT expresses an image as a sum of cosine functions of varying frequencies. It is widely used in image compression.

4.2 Applying DCT

Code Snippet:

```
dct_img = dct2(img);
imshow(log(abs(dct_img)), []);
title('DCT Coefficients');
```

Output Description: Shows DCT coefficients with high-energy components in the top-left corner.

4.3 Inverse DCT

Code Snippet:

```
reconstructed_dct = idct2(dct_img);
imshow(reconstructed_dct);
title('Reconstructed Image from DCT');
```

Output Description: Reconstructs the original image using the inverse DCT.

5. Discrete Wavelet Transform (DWT)

5.1 What is DWT?

Description: DWT provides multi-resolution representation of images using wavelets. It's useful in analysis and compression.

5.2 Applying 2D DWT (Requires Wavelet Package)

Code Snippet:

```
pkg install -forge wavelet;
pkg load wavelet;
[LL, LH, HL, HH] = dwt2(img, 'haar');
subplot(2,2,1), imshow(LL, []), title('Approximation (LL)');
subplot(2,2,2), imshow(LH, []), title('Horizontal Detail (LH)');
subplot(2,2,3), imshow(HL, []), title('Vertical Detail (HL)');
subplot(2,2,4), imshow(HH, []), title('Diagonal Detail (HH)');
```

Output Description: Displays the four subbands of the wavelet decomposition.

5.3 Inverse DWT

Code Snippet:

```
reconstructed_dwt = idwt2(LL, LH, HL, HH, 'haar');
imshow(reconstructed_dwt);
title('Reconstructed Image from DWT');
```

Output Description: Combines subbands to reconstruct the original image.

6. Summary

- **DFT** reveals frequency characteristics of an image.
- **DCT** is efficient for image compression.
- **DWT** gives a multi-resolution view suitable for feature extraction and compression.

Suggested Exercises

1. Retain only low-frequency DCT coefficients and analyze compression.
2. Apply DWT using different wavelet bases (e.g., ‘db2’, ‘sym4’).
3. Remove high-frequency components in the DFT and observe effects on image.