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 8: Multiresolution Image Processing (GNU Octave)

Objective

To understand and implement multiresolution image analysis using techniques like image pyramids and wavelet transforms in GNU Octave.

1. What is Multiresolution Image Processing?

Description: Multiresolution processing involves representing images at multiple levels of resolution or scale. It allows coarse-to-fine analysis, used in compression, object detection, and denoising.

2. Gaussian Pyramid

2.1 What is a Gaussian Pyramid?

Description: A series of images where each level is a progressively blurred and downsampled version of the previous.

2.2 Constructing a Gaussian Pyramid

Code Snippet:

```
pkg load image;
img = im2double(imread('cameraman.tif'));
levels = 4;
g_pyramid = cell(1, levels);
g_pyramid{1} = img;

for i = 2:levels
  g_pyramid{i} = impyramid(g_pyramid{i-1}, 'reduce');
end

figure;
for i = 1:levels
  subplot(1,levels,i);
  imshow(g_pyramid{i});
  title(['Level ', num2str(i)]);
end
```

Output Description: Displays a set of images with decreasing resolution from left to right.

3. Laplacian Pyramid

3.1 What is a Laplacian Pyramid?

Description: A series of bandpass images highlighting the differences between successive Gaussian levels. Useful in compression and reconstruction.

3.2 Constructing a Laplacian Pyramid

Code Snippet:

```
l_pyramid = cell(1, levels-1);
for i = 1:levels-1
  upsampled = impyramid(g_pyramid{i+1}, 'expand');
  upsampled = imresize(upsampled, size(g_pyramid{i}));  % Align sizes
  l_pyramid{i} = g_pyramid{i} - upsampled;
end

figure;
for i = 1:levels-1
  subplot(1,levels-1,i);
  imshow(l_pyramid{i}, []);
  title(['Laplacian Level ', num2str(i)]);
end
```

Output Description: Displays images showing edge-like information across scales.

4. Image Reconstruction from Laplacian Pyramid

Code Snippet:

```
reconstructed = g_pyramid{levels};
for i = levels-1:-1:1
  upsampled = impyramid(reconstructed, 'expand');
  upsampled = imresize(upsampled, size(l_pyramid{i}));
  reconstructed = l_pyramid{i} + upsampled;
end

imshow(reconstructed);
title('Reconstructed Image from Laplacian Pyramid');
```

Output Description: Reconstructs the original image using Laplacian pyramid components. Should closely match the input.

5. Wavelet-based Multiresolution (Alternative)

5.1 Decomposition using DWT

Code Snippet:

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

Output Description: Decomposes image into frequency subbands for analysis and compression.

6. Applications of Multiresolution Processing

- **Image compression** (e.g., JPEG2000)

- **Object detection at multiple scales**
- **Progressive transmission**
- **Edge detection and enhancement**

Suggested Exercises

1. Vary the number of Gaussian pyramid levels and analyze effects.
2. Reconstruct the original image using only some Laplacian levels.
3. Apply wavelet decomposition using different wavelet bases.
4. Use multiresolution techniques to blend two images seamlessly.