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 4: Morphological Image Processing (GNU Octave)

Objective

To introduce students to the basic concepts and operations in morphological image processing using binary and grayscale images in GNU Octave.

1. What is Morphological Image Processing?

Description: Morphological operations are image processing techniques based on shape or structure. They are particularly useful for binary images and are used in tasks like edge detection, noise removal, image enhancement, and object segmentation.

2. Preparing the Environment

2.1 Loading Required Package

Code Snippet:

```
pkg load image;
```

Output Description: Loads the image package, enabling use of morphology-related functions.

2.2 Reading and Binarizing an Image

Code Snippet:

```
img = imread('coins.png');
gray = rgb2gray(img);
binary_img = im2bw(gray, 0.5);
imshow(binary_img);
title('Binary Input Image');
```

Output Description: Displays the binary version of the grayscale image for morphological operations.

3. Structuring Elements

3.1 Creating a Structuring Element

Description: Structuring elements define the shape and size of the neighborhood used in morphological operations.

Code Snippet:

```
se = strel('square', 3);  % 3x3 square structuring element
```

Output Description: Defines a 3x3 square matrix for use in subsequent operations.

4. Basic Morphological Operations

4.1 Erosion

Description: Erosion shrinks objects in a binary image by removing pixels on boundaries.

Code Snippet:

```
eroded_img = imerode(binary_img, se);
imshow(eroded_img);
title('Eroded Image');
```

Output Description: The foreground regions become thinner. Small objects may disappear.

4.2 Dilation

Description: Dilation expands the white regions (foreground) by adding pixels to boundaries.

Code Snippet:

```
dilated_img = imdilate(binary_img, se);
imshow(dilated_img);
title('Dilated Image');
```

Output Description: White areas grow in size. Holes or gaps may be filled.

4.3 Opening

Description: Opening = Erosion followed by Dilation. It removes small objects and smoothens contours.

Code Snippet:

```
opened_img = imopen(binary_img, se);
imshow(opened_img);
title('Opened Image');
```

Output Description: Noise is removed while retaining the shape of larger objects.

4.4 Closing

Description: Closing = Dilation followed by Erosion. It fills small holes and connects disjoint components.

Code Snippet:

```
closed_img = imclose(binary_img, se);
imshow(closed_img);
title('Closed Image');
```

Output Description: Fills small gaps and smoothens object contours.

5. Advanced Morphological Operations

5.1 Morphological Gradient

Description: The gradient is the difference between dilation and erosion. It highlights object edges.

Code Snippet:

```
gradient_img = imdilate(binary_img, se) - imerode(binary_img, se);
imshow(gradient_img);
title('Morphological Gradient');
```

Output Description: Highlights boundaries of objects in the binary image.

5.2 Top-hat Transformation

Description: Extracts small elements or bright regions on a dark background.

Code Snippet:

```
tophat_img = imtophat(binary_img, se);
imshow(tophat_img);
title('Top-hat Transformed Image');
```

Output Description: Displays small, bright objects removed during opening.

5.3 Bottom-hat Transformation

Description: Extracts dark regions on a bright background.

Code Snippet:

```
bothat_img = imbothat(binary_img, se);
imshow(bothat_img);
title('Bottom-hat Transformed Image');
```

Output Description: Displays regions filled during closing.

6. Summary

- **Erosion** and **Dilation** are fundamental operations.
- **Opening** removes noise; **Closing** fills gaps.
- **Gradient**, **Top-hat**, and **Bottom-hat** help extract shape and feature-based information.

Suggested Exercises

1. Use different structuring elements (e.g., disk, diamond) and compare outputs.
2. Apply opening and closing to a noisy image and evaluate effectiveness.
3. Use morphological operations to extract text from an image.