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 11: Image Registration (GNU Octave)

Objective

To understand and implement image registration techniques using GNU Octave. This involves aligning two or more images of the same scene taken at different times, perspectives, or sensors.

1. What is Image Registration?

Description: Image registration is the process of transforming different sets of data into one coordinate system. It is used in remote sensing, medical imaging, and computer vision to overlay images accurately.

2. Preparing Example Images

2.1 Simulating a Transformed Image

Code Snippet:

```
pkg load image;
fixed = im2double(imread('cameraman.tif'));

% Simulate moving image (rotated + translated)
moving = imrotate(fixed, 10, 'bilinear', 'crop');
moving = circshift(moving, [10, 15]);

figure;
subplot(1,2,1); imshow(fixed); title('Fixed Image');
subplot(1,2,2); imshow(moving); title('Moving Image');
```

Output Description: Displays two images with known geometric differences.

3. Manual Control Point Registration (Interactive)

3.1 Selecting Corresponding Points

Code Snippet:

```
[mp, fp] = cpselect(moving, fixed, 'Wait', true);
```

Output Description: Opens an interactive window to manually mark corresponding points between the moving and fixed image.

3.2 Estimating the Geometric Transformation

Code Snippet:

```
tform = fitgeotrans(mp, fp, 'affine');
registered = imwarp(moving, tform, 'OutputView', imref2d(size(fixed)));

imshowpair(fixed, registered, 'montage');
title('Fixed Image (Left) vs Registered Image (Right)');
```

Output Description: Applies the affine transformation and displays the registered result next to the original.

4. Automated Registration using Feature Matching

4.1 Detect Harris Corners

Code Snippet:

```
corners1 = detectHarrisFeatures(fixed);
corners2 = detectHarrisFeatures(moving);
```

(Note: Requires user-implemented or toolbox-based feature detectors)

5. Error Metrics for Evaluation

Code Snippet:

```
error_map = abs(fixed - registered);
imshow(error_map, []);
title('Registration Error Map');
```

Output Description: Visual difference between fixed and registered images.

6. Summary

- Image registration aligns images from different views or times.
- Manual point selection provides precise control.
- Affine transformations handle rotation, translation, and scaling.
- Error maps help evaluate registration quality.

Suggested Exercises

1. Register images using at least 3 manually selected control points.
2. Simulate different transformations (scale, shear) and perform registration.
3. Evaluate accuracy using PSNR or RMSE.
4. Try registering infrared and visible images (if available).