Machine Vision

Morphological Operations and Image Cleaning in MATLAB

Lab Activity Sheet 5

Author: Dr. Mostapha Kalami Heris

Introduction

Why this lab matters

Morphological operations are important tools for cleaning and correcting images after thresholding or segmentation. They help remove noise, fill gaps, smooth boundaries, and separate objects. These steps improve the quality of segmentation and prepare images for later measurement and classification.

What you will learn

By the end of this lab, you should be able to:

- Apply morphological operations such as erosion, dilation, opening, and closing using `imerode`, `imdilate`, `imopen`, and `imclose`.
- Design and choose structuring elements using `strel`.
- Perform background correction using `imopen`, `imsubtract`, and `imtophat`.
- Build a simple image-cleaning workflow that improves segmentation.

Concepts and MATLAB Functions

Morphological Concepts

Morphological operations use a structuring element to probe an image. The shape and size of the structuring element determine how the operation modifies objects in the image. Binary morphology affects object shapes, while grayscale morphology is useful for illumination correction and feature enhancement.

MATLAB Functions

Functions used in this lab include:

- `strel` for creating structuring elements.
- `imerode`, `imdilate`, `imopen`, `imclose` for morphological operations.
- `imtophat` and `imsubtract` for background correction.
- `bwlabel`, `regionprops` for object analysis.
- `imshowpair`, `bwperim` for visualisation.

Dataset and Setup

Use built-in MATLAB images such as `coins.png` or `rice.png`, or use your own images. Always check the image type before processing. Convert images when necessary using `logical`, `uint8`, or `im2double`. Use `imshowpair` or `montage` for comparison.

Safety Checklist and Good Practice

- Confirm the image type using `class` or `whos`.
- Begin with small structuring elements. Large ones can remove objects entirely.
- If background subtraction produces negative values, convert the image first using `im2double`.
- If a morphological operation removes too much detail, reduce the structuring element size.
- Always compare the original image and processed image side-by-side.

Common Mistakes and Tips

- If erosion creates unexpected patterns, your image may not be binary.
- If `imopen` removes the whole object, the structuring element is too large.
- If thresholding fails after correction, normalise intensities using `im2double`.
- If touching objects do not separate, try one or two iterations of erosion instead of a large structuring element.

Activity 1 – Binary Morphology for Cleaning Segmentation Masks

Objective

To clean binary masks by removing noise, smoothing shapes, and adjusting object boundaries.

Key Idea

This activity focuses on applying binary morphological operations with different structuring elements to improve segmentation quality.

Tasks

- Load a binary image (e.g., `coins.png` or a thresholded image using `imbinarize`).
- Create structuring elements with `strel`.
- Apply `imerode` and `imdilate` to observe changes.
- Use `imopen` to remove small noise.
- Use `imclose` to fill small holes.
- Compare results with `imshowpair`.

Checkpoint. After applying `imopen` with a small disk, noise speckles should disappear while main objects stay intact.

Extension and Open-Ended Exploration

Extension: Use directional structuring elements created with `strel('line', length, angle)` to examine effects on elongated features.
Open-Ended Exploration: Design a cleaning pipeline for one of your own images and justify each morphological step.

Activity 2 – Grayscale Morphology and Background Correction

Objective

To correct uneven illumination using grayscale morphology and improve the performance of thresholding.

Key Idea

You will estimate background using a large structuring element and subtract it to achieve uniform brightness.

Tasks

- Load a grayscale image such as `rice.png`.
- Apply global thresholding using `imbinarize`.
- Create a large structuring element using `strel('disk', radius)`.
- Estimate background via `imopen`.
- Subtract the background using `imsubtract`, or apply `imtophat`.
- Threshold the corrected image.
- Compare results using `imshowpair`.

Checkpoint. After background subtraction, objects should appear with more uniform brightness than in the original image.

Extension and Open-Ended Exploration

Extension: Compare open–subtract with `imtophat` using different structuring element sizes.
Open-Ended Exploration: Apply background correction to an image of your choice and explain how the chosen structuring element affects the outcome.

Activity 3 – Object Separation and Morphology-Based Analysis

Objective

To separate touching objects and perform simple object measurements.

Key Idea

Morphological operations can break thin connections so that objects can be labeled and analysed individually.

Tasks

- Load a binary image with touching objects.
- Create a suitable structuring element using `strel`.
- Apply light erosion or `imopen` to separate objects.
- Label objects using `bwlabel`.
- Measure properties using `regionprops`.
- Visualise boundaries using `bwperim`.

Checkpoint. After erosion or opening, previously touching objects should have small visible gaps between them.

Extension and Open-Ended Exploration

Extension: Test different structuring element sizes and observe their effect on object count.
Open-Ended Exploration: Build a full workflow for any image of your choice, including background correction, thresholding, and analysis.

Applications and Further Exploration

Morphological operations are used in many real applications in machine vision and pattern recognition. The ideas that you studied in this lab form the basis of several important tasks in industry and research. Exploring these applications can help you understand how simple operators can solve practical problems.

Example Applications

- **Industrial inspection:** Removing noise, filling gaps, or separating touching components in quality-control systems for objects such as pills, nuts, bolts, and electronic parts.
- **Document processing:** Cleaning scanned text, removing small speckles, connecting broken characters, or extracting lines and shapes from technical drawings.
- **Medical imaging:** Extracting cells, isolating bright structures, removing background illumination, or separating clustered tissue samples.
- **Robotics and automation:** Cleaning segmented masks from depth or camera sensors before object detection or grasping.
- **Environmental and remote sensing:** Highlighting structures such as roads, rivers, or vegetation by removing small noise or enhancing bright features.

Exploring More MATLAB Tools

MATLAB contains many additional morphological and morphological-like functions that you can explore:

- `imreconstruct` for reconstruction-based morphology that can preserve shapes while removing unwanted regions.
- `imfill` for filling holes inside objects.
- `bwmorph` for applying iterative operations such as thinning, skeletonisation, or removal of small branches.
- `bwpropfilt` for selecting objects based on size, shape, or other properties.
- `adaptthresh` combined with morphology for improved local thresholding.

Suggested Directions for Independent Exploration

- Try building a complete workflow for a real image of your choice, such as a biological sample, a circuit board, or text on uneven paper.
- Experiment with different structuring element shapes, such as `strel('rectangle', [m n])` or `strel('line', len, angle)`, and see how they affect specific features.

- Compare global thresholding with local methods and examine how morphology can support or improve each step.
- Explore how combining morphology with filtering (for example, `medfilt2` or `imgaussfilt`) can produce cleaner results.
- Investigate how object measurement changes before and after morphological cleaning.

These ideas can help you connect the techniques from this lab to practical scenarios and encourage you to explore MATLAB functions that go beyond the basic operators used here.

Conclusion and Reflection

This lab introduced the main ideas of morphological processing and showed how these operations help improve segmentation. You explored binary cleaning, illumination correction, and object separation.
Reflect on the following questions:

- Which morphological operations improved your results the most?
- How did structuring element size influence your workflow?
- Which parameters would you expose to a user if you were automating this pipeline?

References and Further Reading

- MATLAB documentation for `strel`, `imopen`, `imtophat`, `bwlabel`, `regionprops`.
- Lecture slides from the morphology session.
- Introductory morphology chapters in image processing textbooks.