STB600 Lab 3: Morphological Image Processing

Atieh Sahraeidolatkhaneh: atieh.sahraeidolatkhaneh@hv.se

Yongcui Mi: yongcui.mi@hv.se

1. Purpose of the Lab

The purpose of this lab is to learn how to extract meaningful information from images using **morphological operations**. You will apply basic operations such as erosion, dilation, opening, closing, and hit-or-miss, and then use morphology to locate and measure a melt pool in an image.

2. Learning Outcomes

After completing this lab, you should be able to:

- apply basic morphological operations in Python using OpenCV;
- understand the effect of structuring element shape and size;
- extract object boundaries using morphology;
- isolate a melt pool region using a morphological pipeline;
- measure the melt pool width, height, and centroid.

3. Start working Environment

Before you start, activate your **stb600** Python environment as in previous labs.

1. Open the **Anaconda PowerShell Prompt**.
2. Move to your Lab 3 folder:

```
cd C:\path\to\lab3
```

3. Activate the environment:

```
conda activate stb600
```

4. General Instructions

During the oral check, you should be able to:

- show the original and processed images,
- explain clearly what each morphological operation does,
- describe why you chose a particular kernel size or shape,
- explain how you isolated the melt pool,
- discuss how you computed width, height, and centroid.

Make sure your code is readable, and always keep your processed images visible.

5. Tasks

This lab contains two main tasks. The first focuses on applying basic morphological operations to a binary image. The second demonstrates how morphology can be used to extract meaningful geometric information from an image.

5.1. Basic Morphological Operations

In this task, you will apply standard morphological operations—erosion, dilation, opening, closing, and the hit-or-miss transform—to the image shown in Figure 1. You should also extract the boundary of the object.

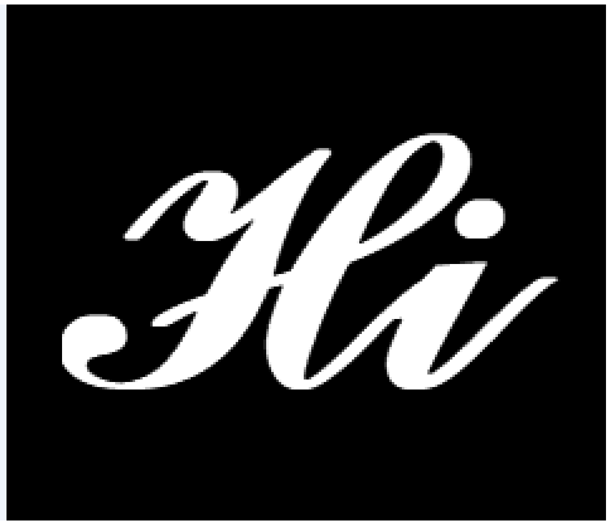

Fig. 1

The following code template illustrates how you may structure your implementation:

```
import cv2

# Read image (binary or grayscale)
# img = cv2.imread("...", cv2.IMREAD_GRAYSCALE)

# Create structuring element
# kernel = cv2.getStructuringElement(shape, ksize)
# Shapes: cv2.MORPH_RECT, cv2.MORPH_ELLIPSE, cv2.MORPH_CROSS

# Apply morphological operations
# img_erosion  = TO DO
# img_dilation = TO DO
# img_opening  = TO DO
# img_closing  = TO DO

# Optional: hit-or-miss transform
# TO DO

# Extract the boundary of the object
# TO DO

# Show or save the results
```

Listing 1: Suggested structure for basic morphological operations

5.2. Object Localization and Size Estimation

In many computer vision applications, it is necessary to identify an object, determine its location, and measure its geometric properties. In this task, you will measure the melt pool width and height, and determine its centroid, using the image shown in Figure 2.

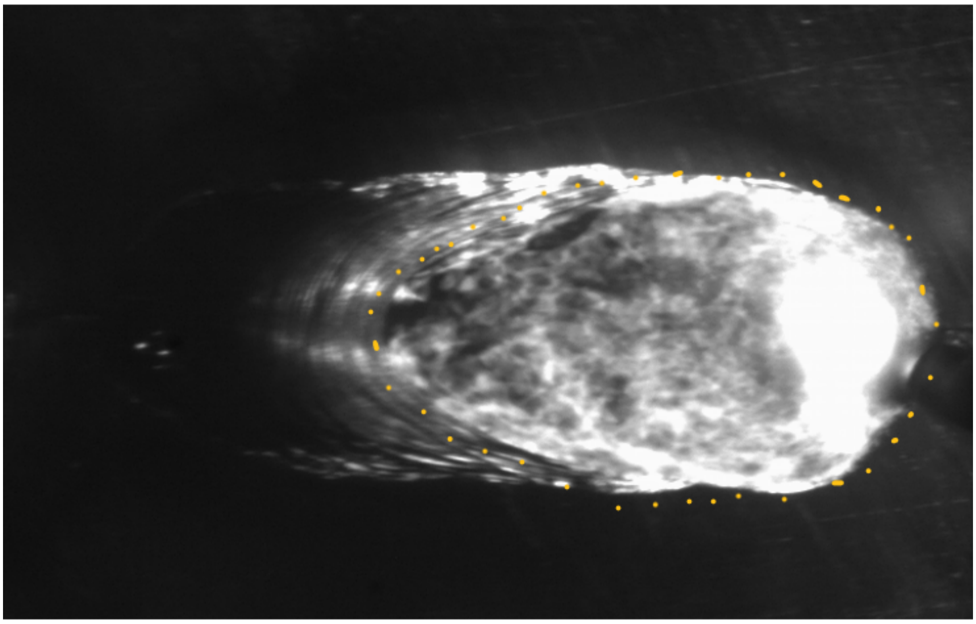

Fig. 2. Melt pool region indicated by the orange spots

A possible workflow is to extract the melt pool region using morphological operations, then identify the extreme points (leftmost, rightmost, topmost, bottommost). These can be used to compute the melt pool width, height, and centroid. You should work on the original image provided.

```
# You may consider thresholding the image, applying suitable
# morphological operations to refine the region, and then
# using the pixel coordinates of the extracted area to
# estimate its size and centroid.
```

6. What You Must Show During the Check

- All morphological results (erosion, dilation, opening, closing, HMT, boundary).
- Melt pool binary segmentation.
- Bounding box and centroid drawn on the image.
- Ability to explain each step clearly.