Interim Progress Report

Priyangika Pitawala

April 16, 2025

Project Title

Image Analysis and Percent Crystallinity Quantification

1 Introduction

This project involves creating a Python-based tool that processes polarized optical microscope (POM) images of a semicrystalline thiol-ene photopolymer system, and analyzes the percent crystallinity. It will also attempt to find a correlation between the percent crystallinity and the thermal and mechanical properties obtained from experimental data.

2 Objectives

- To preprocess the POM images to enhance the contrast between crystalline and amorphous regions, and to reduce noise.
- To define the features that characterize crystallinity and develop an algorithm to distinguish between crystalline and amorphous regions.
- To calculate the percent crystallinity of each image and aggregate the results of multiple images using statistical methods.
- To analyze the relationships between the percent crystallinity and the polymer properties.
- To use data visualization tools to display the results.

3 Methodology

3.1 Key steps and dependencies

Key steps include:

- data loading of polarized optical microscope (POM) images.

- preprocessing to prepare for image segmentation.
- segmentation using edge-based techniques and watershed algorithm.
- crystallinity feature extraction.
- percent crystallinity quantification.
- regression and visualization.

All steps are separated into modules and can be found inside the ‘src/‘ subfolder.

The dependencies are numpy, matplotlib, opencv-python, scikit-image, scipy, and pandas. A comprehensive list can be found in ‘requirements.txt‘ file inside the ‘docs/‘ subfolder.

3.2 Preprocessing requirements

- Grayscale conversion as image segmentation doesn’t require color channels.
- Noise reduction using Gaussian blur and further filtering.
- Masking of known foreground and background.
- Edge detection using Canny operators.
- Gradient magnitude thresholding to highlight crystal boundaries.
- Morphological operations to clean the binary image.
- Computation of segmentation markers.

3.3 Computational strategies

- Separate modules for handling the key steps.
- Interactive exploration, real-time parameter tuning, and segmentation validation using Jupyter notebooks.
- Vectorization of the image processing steps using NumPy and/or SciPy.

3.4 Efficiency and optimization strategies

- Version control by intermittently committing the progress to GitHub, especially prior to tuning parameters to optimize segemntation.
- Image downsampling by removing unnecassary information such as color channels.

4 Project Progress Assessment

I have worked on initial research, data preparation, gradient computation, edge-based segmentation, watershed algorithm, and feature extraction steps. These tasks were supposed to be completed by Week 11 as detailed in my previous Gantt chart. However, the feature extraction step has shown that the image segmentation (and therefore, the preprocessing) step must be optimized further to improve the accuracy. Nevertheless, the current model is providing an acceptable result and framework to qualitatively compare all my POM images.

The updated work schedule is as follows:

- Week 12: Refine image segmentation further; complete extracting the crystallinity features; calculate percent crystallinity.
- Week 13: Regression analysis and model interpretation.
- Week 14: Finalize analysis and data visualization.

No scope adjustment is needed if the segmentation can be refined before the end of Week 12. However, if more time is needed to achieve that, the scope will be adjusted to qualitative rank each image for percent crystallinity based on the model performance, instead of calculating a quantitative number.

The updated gantt chart is as follows:

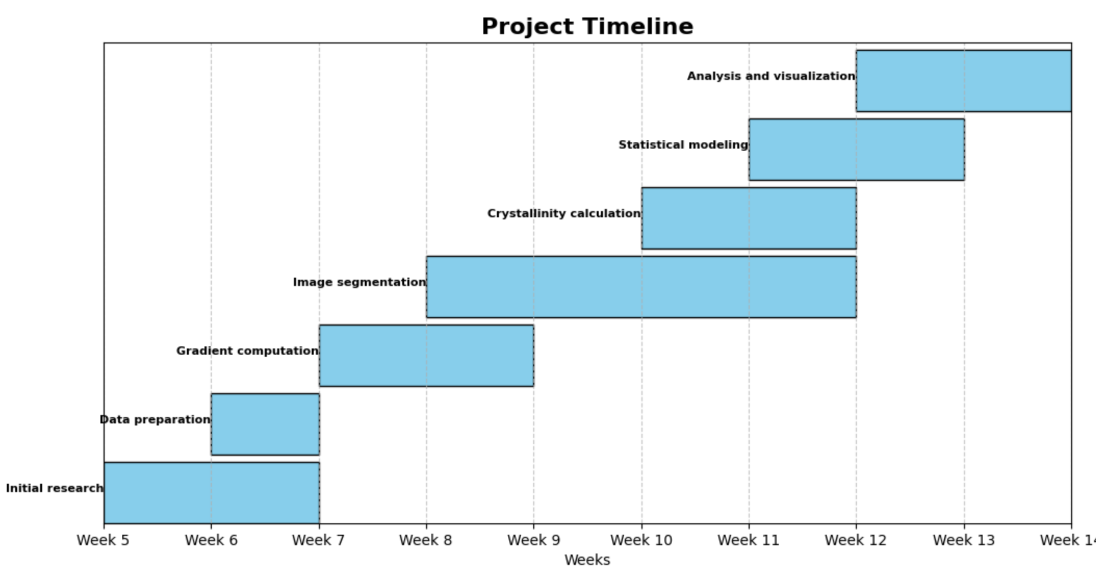

Figure 1: Updated Gantt chart showing planned project timeline.

5 References

1. Gonzalez, R. C.; Woods, R. E. *Digital Image Processing*, 4th ed.; Pearson: Boston, MA, 2018.

2. OpenAI. *ChatGPT*; https://chat.openai.com (accessed March 30, 2025).