

Assignment 1 – Photomosaic

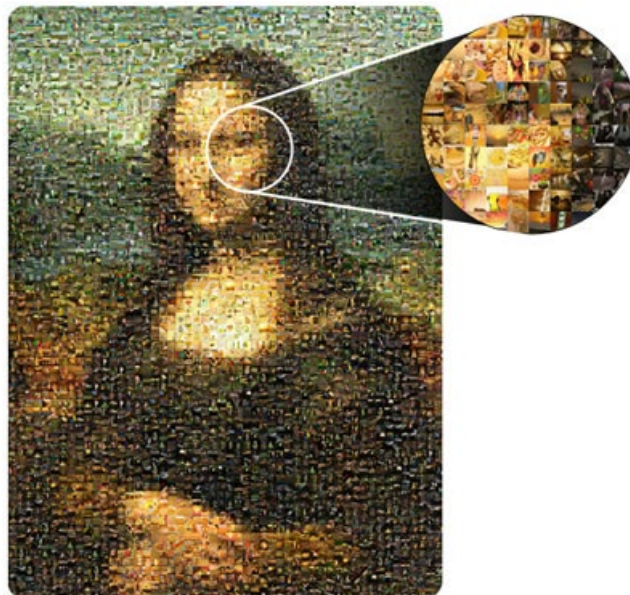
Deadline: Feb. 15th 23:59, submission via Blackboard

Late submission penalty: 10% point deduction per day (maximum 30%)

Plagiarism penalty: course grade **F**

Introduction

Photomosaic is a form of digital art in which a picture is divided into a grid of equal-sized cells, with each cell substituted by a photo of similar colors (similar gray tones in this assignment). It is a popular way of presenting a large number of photos on posters or videos, so that it can tell viewers the overall theme when looked far away, and reveal the content of each single photo in the album when examined closely. In this assignment, you are required to build a photomosaic generator, which creates a grayscale photomosaic of an image with a given set of small images.



General Requirements

- The program should be written in Python3 syntax.

- You are not allowed to use third-party libraries (e.g., *OpenCV*, *NumPy*) in the final version of your *photomosaic.py* and *enhancements.py*, except for *math* library for basic math operations. (The pre-included libraries in *main.py* do not subject to this restriction when they are used for image loading and saving operations). Nevertheless, you are encouraged to use any library to test your program.
- You are required to make sure that the final version of your program is fully operational, especially after you removed extra libraries and commented lines before submission. Each execution failure results in a 10-point deduction.

Basics – Grayscale Photomosaic (80 points)

Your program should be able to process the RGB image into a grayscale photomosaic in a way similar to the following steps:

1. Read the source image and the photo tiles.
2. Use bilinear interpolation to resample the input image to the desired output size and resample the photo tiles to the given cell size.
3. Query for photo tiles with the nearest brightness values.
4. Compose the output image with photo tiles.
5. Save the grayscale output image.

Enhancements (20 points)

You are encouraged to implement some of the following enhancements or other features that you find interesting on top of the standard requirements. Please put your implementations of the enhanced features in a separate standalone source file and name it *enhancements.py*.

Some suggested features:

- Other interpolation methods (e.g., bicubic interpolation) or extrapolation on the borders.
- Other metrics for measuring distances between source image regions and photo tiles, which are perceptually more logical and accurate than mean brightness (e.g., structural similarity). We will accept your method as long as it is correctly

implemented and well explained in your report, no matter how its test-time performance is.

- Dithering methods for approximating an image with large number of colors using a small palette.

Program Structure

Contents in the pack:

- main.py: the main script for photomosaic generation.
- photomosaic.py: the skeleton script in which you are required to implement the core photomosaic functions.
- images/: the folder containing the pictures to be photomosaic-ed.
- tiles/ & tiles_extended/: the folder containing the photo tile files.

Report

You are required to write a report containing the following items:

- Details of each feature.
- Necessary code explanations related to each feature.
- Test-time results.
- Techniques used (if applicable).
- References (if applicable).

Your report is important for our identification and understanding of your implemented features during the grading.

Submission

Deadline: Feb. 15th 23:59.

Format: a .zip file named by your student ID containing:

- report.pdf with everything worth attention, especially about the enhancement part.
- photomosaic.py & main.py & (if applicable) enhancements.py.

All the code should be implemented by yourself. Any kind of plagiarism (including copying online source code or/and code of classmates) in all assignment parts will not be tolerated and will be subjected to disciplinary penalties.