SCC0251 + MAI5020 — Prof. Moacir Ponti

PAEs: Gustavo Evangelista Araújo and Leo Sampaio Ferraz Ribeiro

Assignment 4: Mathematical morphology

Code the assignment by yourself. Ask if you need help. Plagiarism is not tolerated.

1 Introduction

1.1 Goal

The objective of this task is to create and execute the Flood Fill Algorithm for the purpose of painting a specific region and detecting connected components within an image. The algorithm must be capable of producing the pixels belonging to the region's connected components as output.

1.2 Task

1. **Read** the parameters:
   a) **Binary image:** filename of input image.
   b) **Seed pixel x coordinate:** i_k
   c) **Seed pixel y coordinate:** j_k
   d) **Connectivity:** $c \in [4, 8]$
2. **Load binary image:** To load a binary version of the image and avoid compression artifacts, convert it to boolean and back:

```
(imageio.imread(filename) > 127).astype(np.uint8)
```

3. **Apply** the Flood Fill Algorithm using specified connectivity c. Since images are binary, fill colour will always be the opposite of the one in the seed coordinates.
4. **Output** the coordinates (i j) for connected component found. Follow the formatting rules:
   a) Sort the coordinates before displaying, first by i then by j.
   b) Display each coordinate with the (i j) format (parenthesis around, space separated).
   c) Display the list with a single space between each pair: (i_0 j_0) (i_1 j_1) (i_2 j_2)

2 Flood Fill

Extracting connected components from a binary image is essential for many automatic image analysis applications. The connected components of an image are given by the set of pixels neighboring an initial pixel. Given pixel p at coordinate (i_k, j_k), it has four horizontal and vertical neighbors with coordinates: $(i_k + 1, j_k)$, $(i_k, j_k + 1)$, $(i_k - 1, j_k)$, $(i_k, j_k - 1)$. This set of pixels, called the 4-neighborhood of p. The diagonal neighbors of p have coordinates: (i_k+1, j_k+1), (i_k-1, j_k-1), (i_k+1, j_k-1), (i_k-1, j_k+1). Together with the 4-neighborhood, they are called the 8-neighborhood of p [1]. The pixels found by using the neighborhood are the connected components of the specific object within the image.

Flood Fill is a filling algorithm that identifies and modifies the area connected to a specific point in a multidimensional array with a corresponding attribute. It is commonly used in the "bucket fill" tool of painting software, such as the traditional "Paint" program, to fill connected areas of similar colors with a different color. The algorithm is also used in games such as Go and Minesweeper to determine which pieces are cleared [2].

To execute the algorithm, a seed point (a pixel within the area to be filled) is selected. Then, the connected-4 or connected-8 neighborhood is used to fill the area with the specified color. The algorithm searches all nodes in the matrix that are connected to the initial point by a path of the target color and modifies them to the replacement color.

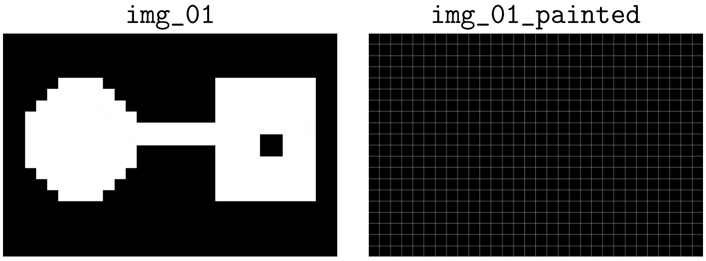

Figure 1: Examples of input and the expected result

3 Input and Output Examples

Input Example 01: One binary image `Dataset/0.tiff`, seed coordinates `(5 5)` and specified connectivity $c = 4$:

```
Dataset/0.tiff
5
5
4
```

Output Example 01: List of coordinates for the connected components found, formatted as specified:

```
(4 4) (4 5) (4 6) (4 7) (5 4) (5 5) (5 6) (5 7) (5 8) (6 4) [continues...]
```

4 Submission

Submit your source code to e-disciplinas (only the `.py` file). You can check for correctness by downloading the test cases from e-disciplinas and testing with run-codes-local, which will be used by the PAEs to grade your work.

1. **Use your USP number as the filename for your code.**
2. **Include a header**. Use a header with name, USP number, course code, year/semester and the title of the assignment. A penalty on the evaluation will be applied if your code is missing the header.
3. **Comment your code**. For any computation that is not obvious from function names and variables, add a comment explaining.

5 Grading

Your work will be graded as:

$$\frac{R+A}{2} - P$$

where each value ranges from $0 - 10$, R is the grade from run-codes-local, A is the grade for the Flood Fill algorithm. P goes up to 1.0 and is a possible penalty for failing to follow the rules from the previous section.

References

[1] R. C. Gonzalez and R. C. Woods. *Processamento digital de imagens .* Pearson Educación, 2009.

[2] A. R. Smith. Tint fill. In *Proceedings of the 6th annual conference on Computer graphics and interactive techniques*, pages 276–283, 1979.