TD4: Morphological operators, connected components extraction, and persistant homology

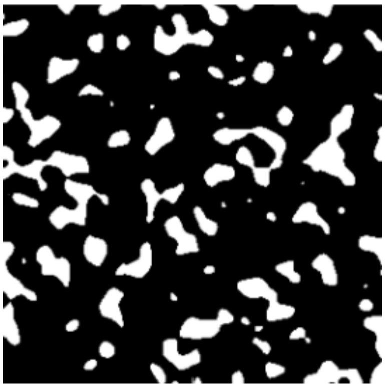

Figure 1: Test image for granulometry applications. The goal is to see the evolution of the number of cells as the structuring element size grows.

In this TP, we focus will develops the tools needed for persistent homology computations. The ingredients needed are:

- **Morphological operators:** The goal is to compute openings as a erosion followed by a dilation. We will focus on binary images.
- **A connected component counter:** The goal is be able to determine the number of connected components in a binary image.
- **Persistent homology:** The goal is to see the evolution of the number of connected components as the structuring element size increases.

Morphological operators

The first tool you will need to implement is the opening morphological operator. This requires implementing the erosion and dilation operators. We recall that, given a binary structuring element S of size $k \times k$ and a grayscale image I, the erosion operator produces an image E such that $E(x,y) = \min_{\{m,n|S(m,n)=1\}}\{I(x+m, y+n)\}$ and the dilation operator D is $D(x,y) = \max_{\{m,n|S(m,n)=1\}}\{I(x+m, y+n)\}$.

Implement the opening operator, which consists in an erosion followed by a dilation.

For a preliminary implementation of persistent homology, plot the curve : $\#whitepixels = f(k)$.

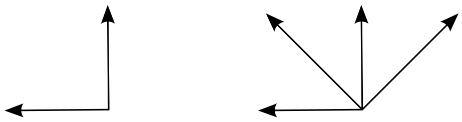

Figure 2: Half-mask for the (1)- and (0)-adjacency.

Exercise 2 Connected components by double-scan

The algorithm consists on a two-pass scan of the image. During the first scan, we identify potential connected components and during the second one, we validate such components using a relabelling step. The main idea is to scan the image starting from top left using half-connectivity masks (cf Figure 2).

In the following, the result of the algorithm will be an image L with one label per connected regions. These labels will be represented by letters. The algorithm is the following

- **First scan:** We apply one of the two half-masks. For each point p:
  - If one of its neighbours q has the same gray level, then we propagate the label ($L(p) = L(q)$).
  - If no neighbour has the same gray level, we create a new label and store it in $L(p)$.
  - If p has several neighbours with the same gray level but with different labels (labels a, b), we pick one of them for p (the "lowest" one, for example) and we keep track that both labels correspond to the same region (i.e. b is a synonym of a) by a so called *Collision list*. For example, you could propagate the lowest label in the neighbourhood and make all labels in this neighbourhood "point" to the lowest one.
- Figure 3 illustrates the overall process with four temporary labels and only two remain after the pruning step. As illustrated in this figure, you could implement the collision list as a list where each node has a pointer to its *lowest* parent during the first scan.
- **Second scan:** Process the collision list such that each label now points to either to itself or to its root (for example, D is now connected to A instead of C). Then, for each pixel we simply remap the labels according to the pruned collision list.

Questions:

1. Implement the algorithm described above, for the the $(0)-$ and the $(1)-$adjacencies. Use this algorithm on the provided input png file, then output the label map in png format.
2. Ouput the number of connected components

Exercise 3 Persistent homology

We are now ready to compute persistent homology and see how the number of connected components of our image varies when increasing the morphological structuring element size. Progressively increase the size of the structuring element and plot the number of connected components as a function of the structuring element size.

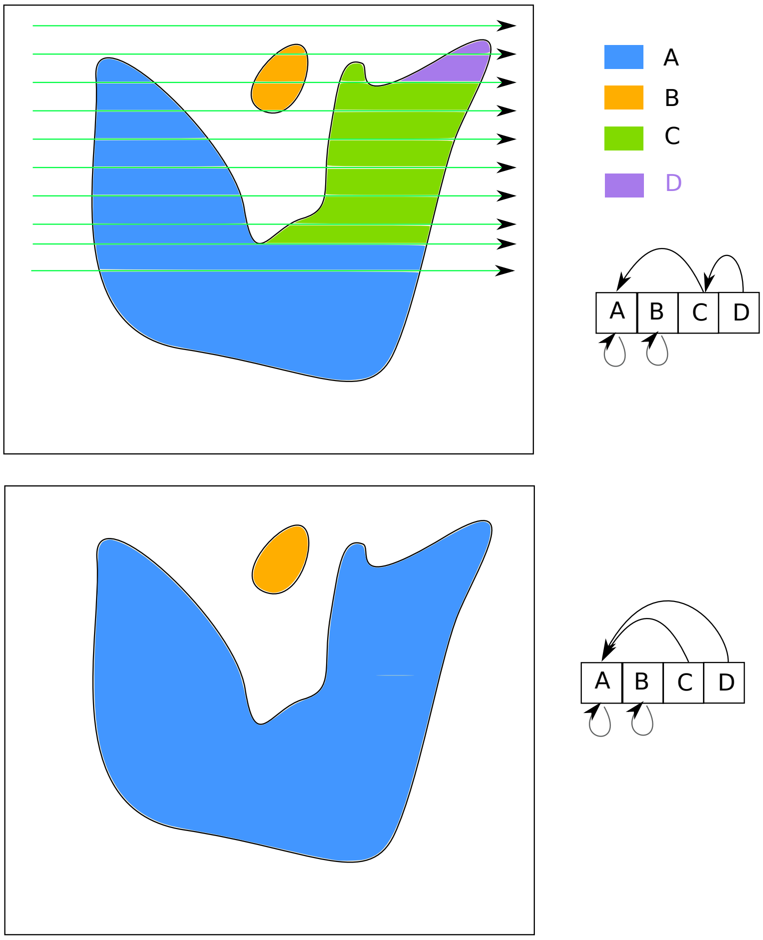

Figure 3: Result of the first pass and relabelling.