TD6: Border Extraction and DSS Segmentation

In this TP, we perform elementary geometrical analysis of digital contours using Digital Straight Segments. We assume that:

- You have DGtal and DGtalSkel tools up and running (cf TP5)
- You have the function that construct the Gauss digitization of an Euclidean disc (as a `DigitalSet`, cf TP5)

Exercise 1 Border extraction (sequel and end)

The objective here is to construct the ordered sequence of pixels (either (0)- or (1)-connected) corresponding to a $(\kappa, \lambda)-$Jordan pair. The algorithm is the following:

- Scan the domain to locate a first border point
- Given an orientation (cf Figure 1) and the last two points, check the neighbors points to decide the next border point. More precisely, starting by the first point, if it belongs to the object, it is the next one. Otherwise, "probe" the next one.

At each step, you'll have to "rotate" the half-masks according to the last move. You can easily check that the above algorithm generates points which are $\lambda-$adjacent to background points.

Question Implement such border tracker and test it on the `DigitalSet` obtained by the Gauss digitization of an Euclidean disc. Note that the output of the tracker cannot be a `DigitalSet` (which is an un-ordered set of points). Instead, you could use a `vector<Point>` (don't forget the `using namespace std;`).

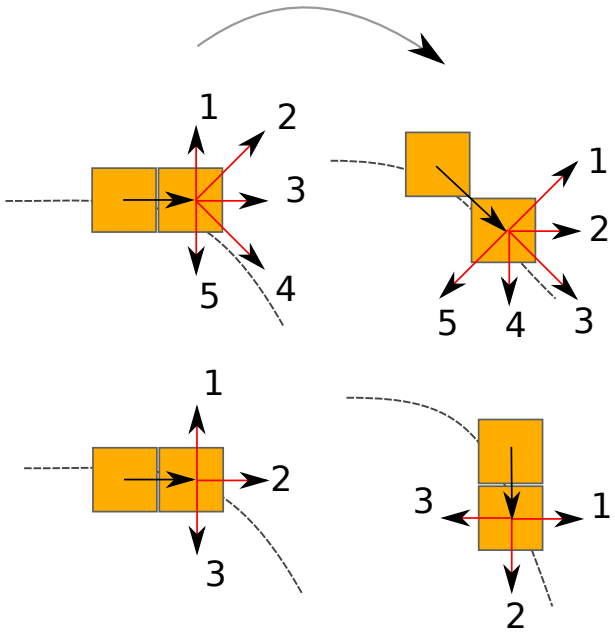

Figure 1: Illustration of the border tracker algorithm.

Exercise 2 Segmentation into maximal DSS

In `DGtal`, we use the DSS recognition algorithm implemented in the `ArithmeticalDSS` class. In the DGtalSkel folder (checkout the last version), you would find and example of `ArithmeticalDSS` usage. Note that the technical documentation of this class is available at `http://libdgtal.org/doc/stable/classDGtal_1_1ArithmeticalDSSComputer.html`.

Questions:

- Have a look to the `DSSExample.cpp` file. Use `front()` and `back()` methods and return the Euclidean length of the DSS.
- Customize `DSSExample.cpp` to add more points (such that the complete sequence is not a DSS anymore) and construct a decomposition of such contour into maximal DSS (cf Figure 2 for an example for the 4-connected case).

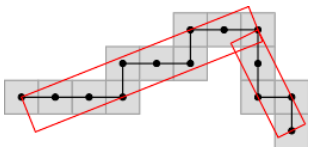

Figure 2: Example of a 4-DSS decomposition of a small point sequence.

- From the first exercise, compute the sequence defined as the border of a digital disc and compute its decomposition into maximal DSS. Display the contour and each segment to a board output and estimate the length of the contour in the meantime (sum of DSS Euclidean length). Compare the estimated lengths to $2\pi R$ when you increase the ball radius.