DIGITAL GEOMETRY (UPEM MASTER 2 INFORMATIQUE “IMAGERY SCIENCES”)

Laboratory Session 1:

Pixelation of Euclidean shapes

Yukiko Kenmochi

November 13th, 2020

Practical information

Submission: Upload a compressed file containing your code and report with the file name of "your last name" via eLearning.

Deadline: November 24th, 2020

Evaluation environment: Linux Ubuntu (This means that your program will be compiled and ran in a Linux environment for the evaluation, without any special setting.)

Note: Please include with your code a CMakeLists.txt file, which facilitates its compilation, and a Readme file, which gives examples of command lines. Please do not include files which are not necessary e.g., binary files obtained from a compilation of the code.

Preparation

INSTALL DGTAL

You need Digital Geometry Tools and Algorithms Library, DGtal for the laboratory sessions. On the PCs in the laboratory room, the library is not installed by default. Thus you need to settle the working environment where DGtal is pre-installed. There are two ways to achieve this.

You install DGtal in your directory of a laboratory room PC (or on your personal PC), for which either Linux or Mac OS is recommended:

1. download the package here, and
2. install DGtal following the instruction here.

Please also see here for quick install. As only 2D images are treated, no particular install option is required except for CAIRO.

If you encounter some difficulty in installing DGtal, you can also download the Linux virtual machine (ubuntu 18.04) where DGtal (version 1.1) is pre-installed. The image file is availbale here (attention to the file size, 3.6GB!), and play this with VMware player (if this is not yet settled on a PC, please install it). Once your virtual machine is running, you can use the account whose user id is "user" and password is "dgtp".

Generate your project using CMake tools

It is recommended to use CMake tools for your project as your program needs to be associated to DGtal with all its dependencies. In your project directory, please make a CMakeLists.txt file, following the instruction here. You can also download an example here.

If you can compile a simple program such as the example *helloworld.cpp* (see here) with CMake, for example:

```
mkdir build
cd build
cmake ..
make
```

then, you are ready to start the following experiments.

Experiments

Step 1: Define Euclidean shapes with implicit functions

We consider Euclidean shapes as compact (i.e., bounded and closed) sets in $\mathbb{R}^2$, which are defined by implicit functions. For example, a disk of center (c_x, c_y) with radius r is defined by

$$\{(x, y) \in \mathbb{R}^2 : f(x, y) \leq 0\} \tag{1}$$

where $f(x, y) = (x - c_x)^2 + (y - c_y)^2 - r^2$. In the same way, we can also define an eclipse including interior points.

Can we define a convex polygon using implicit functions?

Can we exactly calculate the area and perimeter of such Euclidean shapes defined by implicit functions? If so, please give their formulas in the report.

Step 2: Discretize Euclidean shapes and extract digital shape boundary

Download the program, main.cpp, to

1. test Gauss discretization of a disk (see here for the explanation of the module of "Shapes, Shapers and Digitizers").
2. extract the inter-pixel boundary. Please understand the detailed steps as follows:
   a) make a cubical complex from a "digital object" corresponding to each connected component. For that, we first create a topological space made from grid cells, which is called a Khalimsky space in DGtal, and initialize the space from the "digital object" (see here for the more information).

b) Extract the boundary of each connected component as a set (sequence) of 1-cells (see here for the practical information).
c) Use Board2D to visualize the inter-pixel boundaries (show such figures in your report).

Understand the program with a help of associated documents, and test it by varying shapes (ellipse, triangle, square, etc.) and sizes (or image resolutions). If you notice something, please make remarks in the report.

STEP 3: COMPUTE AREA AND PERIMETER BY COUNTING CELLS

For each generated digitized shape, calculate its area and perimeter as the number of the 2-cells, which is equal to the number of grid points in a "digital object", and the number of 1-cell of the boundary. Compare them with the ground truths that are analytically calculated from an original Euclidean shape.

Make the same numerical experiments to discretization results with different image resolutions, and verify the multigrid convergence for each geometric measurement by drawing a diagram (resolutions vs errors). Please show such diagrams (with various shapes) in the report and make an analysis and/or a discussion on them.

STEP 4: MAKE THE CONVEX HULL OF DIGITAL SHAPE BOUNDARY

Download the program, convexHull.cpp, which make a convex hull from a given set of points. Modify this program to obtain the convex hull of each digital shape boundary, which was previously generated. Please visulize the results and show some figures in your report. Is this a good approximation of any digital shape?

STEP 5: COMPUTE AREA AND PERIMETER VIA CONVEX HULL

As an alternative approximation of area and perimeter measurements to that of Step 4, please calculate the area and perimeter of the convex hull (generated in Step 5), for each digitized shape (generated in Step 3). Similarly to Step 4, compare them with the ground truths that are analytically calculated from an original Euclidean shape, and make the same numerical experiments to discretization results with different image resolutions. Verify then the multigrid convergence for each geometric measurement by drawing a diagram. Please show these diagrams in the report and make an analysis and/or a discussion, compared with the results in Step 4.

STEP 6 (OPTIONAL): EXPERIMENTS WITH 3D EUCLIDEAN SHAPES

Do Steps 2 and 3 for 3D Euclidean shapes: discretize 3D Euclidean shapes and extract their inter-voxel boundary (Step 2), and calculate the volume and surface area by counting 3- and 2-cells, respectively (Step 3). Please show experimental results and give your comparative study of 2D and 3D in the report.