

Exercise Sheet 1 - CONRAD Basics

Andreas Maier, Jennifer Maier, Bastian Bier,
Alexander Preuhs, Christopher Syben

October 17, 2017

This course builds on the open source software framework CONRAD. In this exercise, we make ourselves familiar with the basics of the framework and do some fundamental operations.

1. **Installation/GitHub:** Follow the CONRAD installation guide¹ and note that:
If you are working on the cip machines, clone the repository to the following directory, that you will also set as Eclipse workspace (instead of `C:/Reconstruction`): `/proj/i5fpctr/YOUR_DIRECTORY/`
2. **Personal Working Directory:** Once installed, create a new package with your custom group name inside the **tutorial** tree of CONRAD. All your code goes here.
3. **Creating a Grid2D:** During the course, you will be working with CONRAD's numeric `Grid2D`.
 - (a) In your project folder, create a new Class `LearnGrid2D.java`. In the `main` method, create a new instance of the `ImageJ` user interface², which you can later use to perform operations on your grid.
 - (b) Now, create a new `Grid2D` object (package `conrad.data.numeric`) of arbitrary width and height. Choose and set the spacing (different values for both dimensions), and compute and set the origin of the grid as described in the lecture slides.
 - (c) The method `setAtIndex(int i, int j, float val)` of the class `Grid2D` can be used to set the value of a pixel at index `(i,j)` to a certain value `val`. Create a static method `void drawRectangle(Grid2D grid, int xC, int yC, int a, int b, float val)`, that draws a rectangle of pixel size `(a,b)` around the pixel coordinates `(xC, yC)` filled with the value `val`. Draw some rectangles of different size and value in your grid and use the method `show()` of `Grid2D` to have a look at your grid.

¹<https://www5.cs.fau.de/lectures/ws-1718/projekt-flat-panel-ct-reconstruction-projcr/exercises/>

²<http://rsb.info.nih.gov/ij/docs/>

4. **Accessing data in a Grid2D:** Now that you created your `Grid2D`, let's play around with it a bit.
 - (a) `Grid2D` has a method `getAtIndex(double i, double j)`, that can be used to access values in pixel coordinates. Since you have set the spacing and origin of your `Grid2D`, you are able to convert between world coordinates and pixel coordinates using `Grid2D`'s methods `indexToPhysical()` and `physicalToIndex()`. Create a method `double[] getAtPhysical(Grid2D grid, double x, double y)` to access a value in your `Grid2D` at world coordinates. What problem can occur when accessing the `Grid2D` values this way?
 - (b) The next thing we want to do is to read a value from our `Grid2D` at non-integer **pixel** coordinates. For this, you can not use the method `getAtIndex()` anymore. The class `InterpolationOperators` will help you with this task. Please make sure that you understand the difference between what you did before (access at world coordinates) and what we want to do now!
5. **Executable class:** Create a Java class `Executable.java`. In the `main` method of this class, you will execute all the algorithms you implement during this course. For now, only create an instance of `ImageJ`.
6. **MyPhantom:**
 - (a) Create a Java class `MyPhantom.java` that is derived from the `Grid2D` class and design your custom phantom. It is a 2D image of **variable size and spacing** that you can use later to test your algorithms. Set the input to the Constructor accordingly.
 - (b) Your phantom should consist of at least three geometric objects with different intensity values. You can copy your method `drawRectangle()` from the previous task. Change it, so that you can call it on an instance of `MyPhantom` instead of giving the grid as input parameter. Think of two other geometric objects and implement those as well.
 - (c) In the `main()` method of the class `Executable`, create an instance of your phantom and use `show()` to look at it. Use your phantom to get familiar with `ImageJ`: How can you apply math operations, measure a line profile, set an ROI or quickly determine mean, minimum and maximum value?

Checklist:

- CONRAD up-to-date and running
- Your package committed to your own branch
- Know how to use `Grid2D` and world/pixel coordinates
- Created your own 2D phantom
- Validated by a supervisor