Exercise- A pipeline for measuring the interface area of a structure in a 3D image

In this exercise we will make a simple pipeline for measuring the interface area of a structure, from a 3D image of the structure. The structure is made by freeze casting and is used for heat transfer in magnetic refrigeration. The structure has two phases: pore (low intensity) and solid (high intensity) We are interested in measuring the volume specific interface area of the structure. The 3D images we will be working with were obtained from an x-ray scanner at the 3D imaging center at DTU. The full dataset is 1000^3 voxels, this exercise uses a smaller subset to reduce the memory requirements and the processing time.

If you want to know more about freeze casting: https://en.wikipedia.org/wiki/Freeze-casting

Exercise structure

Each step of the exercise has a corresponding section in the Exercise_start.m script in a separate section (between two '%%').

Each step has a list of TASKS that you should perform and some HINTS that will help you accomplish them. Often the hints are in the script as example code or comments.

Implement your code directly in the supplied Exercise_start.m script unless stated that you should implement a function.

Store modified data in new variables to be able to make comparisons with previous steps and to avoid having to re-run previous sections. E.g. if you create a filtered image store it in a new variable.

Some of the hints prints to the Command Window. You can suppress this by ending the line with ';'. Alternatively, you can comment out the line with a preceding '%' or by highlighting some code and pressing ctrl-r (ctrl-t to remove the comment).

Tips on working with Matlab

It is recommended to not run the entire script as this will result in errors because the script is not finished. Instead, while developing the script it is recommended to only run a section or a few lines at a time.

To execute a single line of code you can highlight the line or lines and press F9 (on windows). This is equivalent to typing the highlighted text in the command window. This is also useful if you want to compare two specific plots. Just highlight and run the desired plot commands.

To run one section of code at a time (everything between a set of '%%'). Press ctrl-Enter to execute a single section or press the Run section button in the GUI.

To interrupt Matlab (e.g. if you accidentally print the full image to the command window): press ctrl-c.

To get help on a specific Matlab function, type “help function_name” in the Command Window

```
help histogram
```

You might be prompted for a missing toolbox. If so just follow the link that shows up with the message of the missing toolbox to install it.

Exercise 1 – Load and show the data

The dataset consist of a 500x500x500 voxel 3D image. Depending on your computer and your patience it is advised to only use a smaller part of the full dataset. The code in the script loads the entire dataset but only retains a 300x300x300 sub volume for further calculations.

TASKS:

- Run the “load in the data” section in the script, you will be prompted to select the folder where you saved the image data (The “freezecast_reduced_8bit” folder).
- Change the slice number in the code to show images of other slices in the 3D image. You can use a for-loop if you want to show all the slices of the volume in sequence.

Exercise 2 – filter the data

To reduce the noise we typically filter the data before segmenting it.

TASKS:

- Filter the 3D image stored in the variable vol using a median filter. A small filter should be sufficient for this dataset ([3 3 3]). Store the filtered image in a new variable for convenience.
- Plot the filtered image as above and compare it to the raw image data. Does it look less noisy?
- Plot the histogram of the filtered image and compare it to the raw. How has the histogram changed?

Exercise 3 – Segment the data

To prepare the data for quantitative characterization we first segment the data into the pores and the solid phase.

TASKS:

- Determine a threshold value. e.g. halfway between the peaks for the pore and the solid phase in the histogram.
- Apply the threshold to obtain a binarized 3D image with values of false (0) for the pore phase and true (1) for the solid phase. Store the segmentation in a new variable.
- Plot slices of your segmentation and compare it to the image data to convince yourself that the segmentation captures the underlying structure of the image.
- Segment the raw image data (not filtered) and compare it to the segmentation of the filtered data. How does the pre-processing affect the segmentation?

Exercise 4 – Calculate phase fractions

Now that we have a segmentation, it is time to extract some parameters that describe the segmentation. We start with the most basic, which is just to find the fraction of each phase. I.e. what percentage of the imaged volume is taken up by pore and what percentage is taken up by solid.

TASKS:

- Create a function in a new file that calculates the phase fraction of the phase represented by 1 (true) in the segmentation. An empty function to edit is supplied in the file get_volume_fraction.m.
- Test your function by making sure that the phase fraction of the pores and the solid phase sums to 1.

Exercise 5 – Polygonize the interface

We use a built-in Matlab function for polygonizing an interface (isosurface). We will shortly use this polygon description of the interface to visualize the 3D structure and to calculate the interface area. This function returns a list of triangles that best describes the interface defined by a certain iso level in the 3D image. If you are interested in how this works, take a look here:
https://en.wikipedia.org/wiki/Marching_cubes

These types of methods are investigated further in course 02580 Geometric Data Analysis and Processing.

We chose the iso level at 0.5 since the two phases in our segmented data have values 0 (pore) and 1 (solid). I.e. we are asking the algorithm to extract the surface that separates these two phases.

TASKS:

- Change variable names in the supplied code to match your chosen variable name for the segmented image.
- Understand the structure of the *fv* variable. It describes all the vertices and faces of the polygons using an indexed face set data structure (see lecture slide for explanation). You will need this shortly.

Exercise 6 – Make a visualization of the pore-solid interface

Here we just just draw the polygons that we obtained above to visualize the structure.

TASKS:

- Change variable names if needed. You can rotate the visualization by using the “Rotate 3D” tool that appears when you hover the mouse over the figure.

Exercise 7 – Calculate the interface area

Here we exploit the polygon data we created above to create a function that calculates the interface area.

TASKS:

- Implement a function that calculates the interface area of a segmentation. A template function is supplied (get_interface_area.m).
- Convert the coordinates from voxel units (image voxel positions) to physical units through the supplied voxel size. HINT: the initial unit of the vertex positions is [voxel].

Exercise 8 – Test the interface area function

Now we test the interface area function by applying it to a dataset where we know what the interface area is. In this case, an artificially created sphere.

The supplied function (get_sphere) creates a segmentation of a sphere with a specific radius (in voxel units). The output variable sphere_interface_area is in units of voxels.

Note, since the segmented sphere has a rather blocky surface you should expect your calculated interface area to be approx. 8% larger than the theoretical sphere_interface_area.

TASKS:

- Test your function by calculating the interface area of the variable vol_sphere (it should be approx. 8% larger).
- Use the function smooth3 to smooth the segmentation of the sphere before calculating the interface area. How does this affect the accuracy? How close do you get to the theoretical value?
- Increase the voxel size to [2 2 2]. How much does your interface area increase? Is this as expected?

Exercise 9 – Interface area of the freeze cast structure

Now that we have a validated function for measuring the interface area we apply it to the 3D image of the freeze cast sample.

TASKS:

- Use your function to calculate the interface area of the freeze cast structure in physical units.
- Convert this to the volume specific interface area. i.e. the interface area per volume unit. What is the unit of this value?

Exercise 10 - Measurements as a function of volume size

To get measurements that are independent on which part of the structure we measure on we need a certain representative volume size. Here we explore when the volume size is sufficient.

TASKS:

- Make plots of the volume specific interface area as a function of volume size. Some starting code is supplied in the script.
- How much do the values vary as a function of volume size? Why do you think that is?