Segmentation of 3-D glass material images: from raw data to physical measurements

Chloe Brillatz
Surface du Verre et Interfaces
15/11/2018
The motivations

barium borosilicate glass
$\text{SiO}_2 + \text{BaO} + \text{B}_2\text{O}_3$

128 microns
The motivations

Thin layer

Main glass

Nano-texture

Super hydrophobic glass
From raw glass to physical measurements
Data acquisition

3D imaging by X-ray tomography at the Grenoble Synchrotron (European Synchrotron Radiation Facility):

- Non-destructive method + evolution in time
- 900°C
- Nanometric resolution (ESRF Beam 16b)
Data: horizontal view

0.0 s

4371.0 s

9146.0 s

128 microns

108 microns

Full glass sample

slice view

Movie
The goal
After local filter

ndimage.median_filter from scipy
- Computes the median of the voxel values in a given window, the central voxel is replaced with this median value

Before

After
threshold_otsu from skimage.filters
- Reduces a gray level image to a binary image
- Globally calculates the threshold
After segmentation – 2D view

- rank.otsu
  from skimage.filters
  - Reduces a gray level image to a binary image
  - locally calculate the threshold
The goal
Some numbers

Time: 0 s
1.8 billions voxels
7 GB

x 80

Time: 4 hours
Total 560 GB

Parallel from joblib

50 chunks
overlap = 10 slices
30 cores

Total running time for the segmentation: 17 hours
After segmentation – 3D view

- `marching_cubes_lewiner` from `skimage.measure`
- `finds surfaces in 3d volumetric data`
- `mlab.triangular_mesh` from `mayavi`
After segmentation – 3D view

```
marching_cubes_lewiner from skimage.measure
- finds surfaces in 3d volumetric data
mlab.triangular_mesh from mayavi
```
Labelling

label from scipy
morphology.remove_small_objects
from skimage
Data analysis

Evolution of the volume fraction over time (on masked data to avoid the reconstruction border effects)

ba1270air Volume distribution -

volume (vox number = 30-3 micrometers³)

% of objects per million

time (in seconds)
After segmentation – 4D view

**TimeVisualizer - Mayavi**
- A GUI for 3-D visualization and exploration of a time series of 3-D images
What to take home

- start simple
- compromise between quality and time
- Python simple and powerful tools

1 TB
I’d be happy to share more details, the codes and ideas:
Please contact me: chloe.brillatz@gmail.com