StrutSurf v1.0

Reece Oosterbeek
Department of Mechanical Engineering, Imperial College London

October 7, 2021

---

1 Introduction

StrutSurf is a program designed for analysis of micro-CT images detailing the surface of struts within 3D-printed lattice structures. Built using MATLAB, StrutSurf makes use of inbuilt functions, and the iso2mesh toolbox developed by Qianqian Fang et al. [1, 2].

Generating a mesh with the resolution required for surface analysis, across an entire micro-CT scan volume would result in prohibitively large file sizes. StrutSurf therefore allows users to generate a smaller, high resolution mesh, around selected points within the micro-CT volume. Individual struts can then be selected and processed for analysis, including position, elevation angle, strut diameter (semi-major and semi-minor diameters for elliptical struts, minimum and maximum Feret diameters), *Sa*, *Sq*, and *Sz*.

Processed struts and surfaces can be exported as .stl meshes or XYZ coordinates in a .csv file. Analysis results are saved to a .csv file along with processing parameters, to enable reproduction of results without needing to save large analysis files.

Requirements: StrutSurf requires installation of MATLAB Runtime, version 9.9 (MATLAB R2020b) or later. Available at `https://uk.mathworks.com/products/compiler/matlab-runtime.html`. If not already installed, installer will download and install the most recent version.

Important: All parameters set and measurements obtained in StrutSurf are in pixel units, except rotation and elevation angles specified with a ° symbol. These can easily be converted to physical units by multiplying by the pixel size.

All images and plots displayed in StrutSurf use UIAxes, which contains a toolbar () that allows zooming, panning, view reset, and saving of the current view.

2 Importing Data

StrutSurf is designed to import micro-CT data in the form of binary image slices, in .bmp, .tif, .tiff, .jpg, .png formats. This is carried out in the **Import tab**.

Requirements: StrutSurf requires binary (black and white) images only, not greyscale. To convert greyscale images to binary, ImageJ (Fiji) is recommended `https://imagej.net/Fiji`.

Browse files: will open a dialog box where the user can navigate to the images to be analysed. Select one of the image slices in the folder for analysis and click open, and StrutSurf will populate a list of files to import based on the file name. Clicking these files will load a preview under **Raw Image**.

Import files: will load the list of files into StrutSurf. This can take several minutes. Auto-cropping is carried out during this process unless deselected using the **Auto-crop during import** checkbox. This process defines a bounding box around detected white pixels and adds a small buffer (defined by the **Buffer Pixels** parameter) to define the crop region. If files are imported without auto-cropping, this can be run later using the **Run auto-crop manually** button.

After importing, a combination of all image slices in the XY plane will be displayed under **Crop Shadow**. The crop region will be displayed using a red border, and the centroid is displayed with an "×" symbol. The imported data can viewed along X, Y, and Z planes in the **Imported Slices** panel and the scroll bar along the bottom.

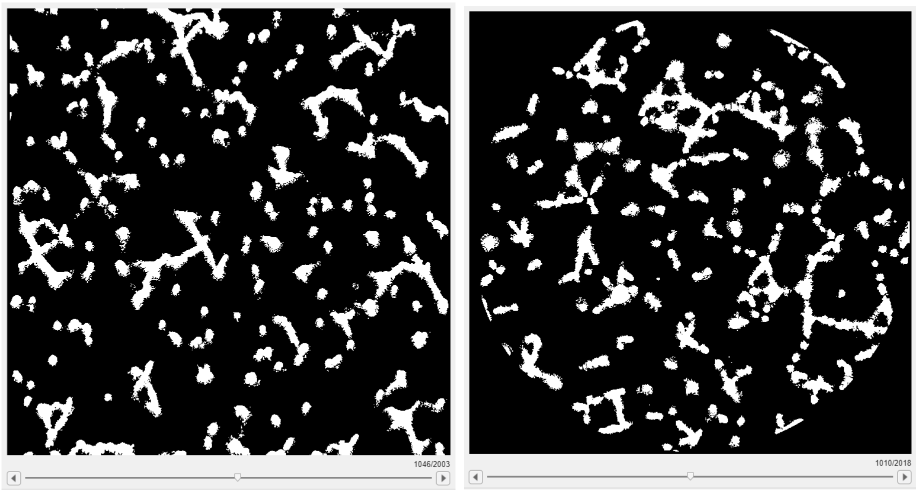

Figure 1: Imported micro-CT data of a strut-based lattice viewed along Y (left) and Z (right) axes.

3 Region Selection

Once micro-CT data has been imported, StrutSurf can be used to generate a mesh of a reduced sample region around a selected point. This is carried out in the **Sample Region** tab.

The size of this region can be set by the **Sample Region Size** parameter, and is generated centred around a point that is manually defined, or randomly generated. Manually adding a point can be useful for recreating previous analyses or sampling at systematic points, and is done by entering pixel XYZ pixel values and clicking **Manually add point**. Selecting the correct region size requires some trial and error, as there is a tradeoff between keeping the region small to speed up processing and make it easier to see and position the mask, and ensuring it is large enough to have a good chance of containing a full strut. Double the estimated length of a strut (converted to pixel units) is a reasonable starting value.

Add random point adds a random point within the image volume, ensuring that the sample region will not exceed the volume boundary. Random points can be generated using a selection of algorithms:

- **Cylinder: Radially uniform:** Generates random points in a cylindrical volume centred at the centre of the cropped volume. Cylinder circular cross-section lies in XY plane. Sampling is uniformly distributed with respect to r - the radial distance from the centre.
- **Cylinder: Cartesian uniform:** Generates random points in a cylindrical volume centred at the centre of the cropped volume. Cylinder circular cross-section lies in XY plane. Sampling is uniformly distributed within the cylinder across the XY plane.

- **Cartesian uniform:** Generates random points across the XY plane, with a square rather than circular cross-section.

Results from these algorithms are shown in Fig. 2. A higher sampling density is seen at the centre for the Cylinder: Radially uniform algorithm to ensure consistent sampling over r. All algorithms show a gap between the edges of the volume and the sampled points, to ensure the generated region does not exceed the volume bounds.

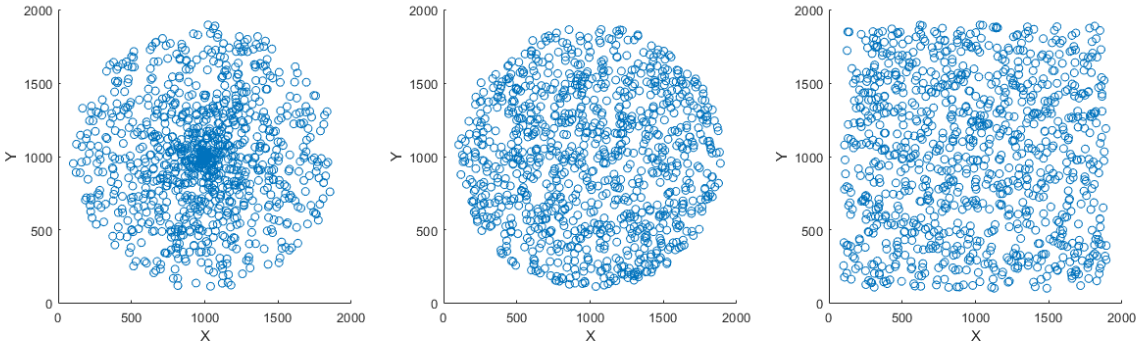

Figure 2: Points generated by the Cylinder: Radially uniform (left), Cylinder: Cartesian uniform (middle), and Cartesian uniform (right) algorithms, in the XY plane. 1000 points generated in a 2000x2000 pixel plane, with a region size of 200.

To generate a sub-region mesh, select the desired point from the list, and click **Generate sub-region**. The XYZ co-ordinates displayed next to this button show the selected point that will be used for this sub-region A mesh will be generated and displayed in the **Sub-region mesh** panel, and the XYZ co-ordinates for this region are displayed above the viewing panel. During mesh generation, small disconnected volumes (islands) can be filtered out by checking the **Remove disconnected objects** checkbox and entering a size threshold. Isolated objects containing fewer than this number of voxels will be removed as demonstrated in Fig. 3. This can help remove unconnected particles or image artefacts that can significantly impact surface roughness measurements, particularly Sz.

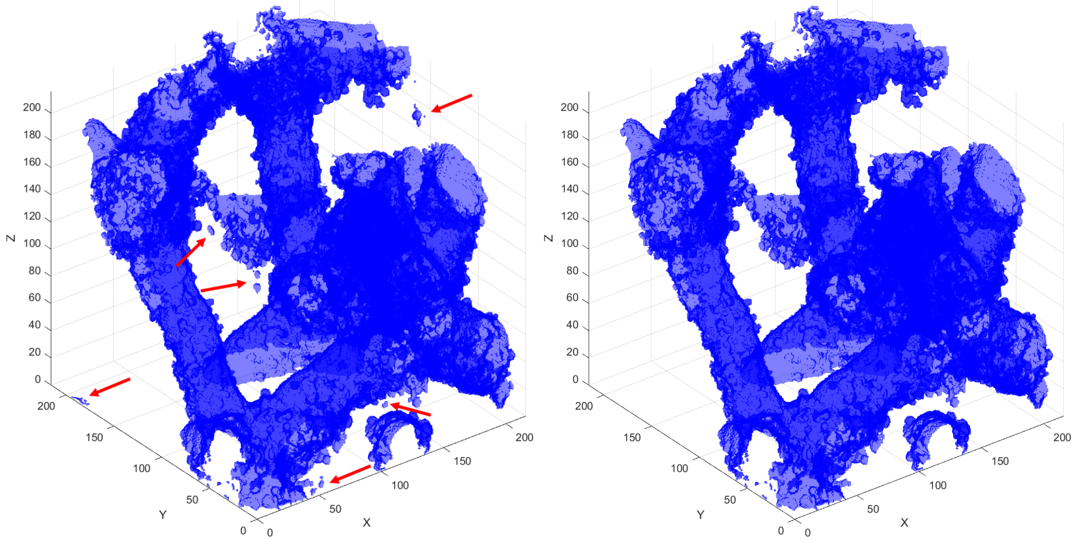

Figure 3: Sample sub-region mesh, without removing disconnected objects (left) and removing disconnected objects <500 voxels.

This mesh can be exported to an .stl file using the **Export stl mesh** button.

4 Masking struts

Once a mesh for a sub-region has been generated, in order to perform detailed strut analysis (diameter, angle, surface quality etc.) a single strut must be selected. This is done in the **Mask** tab, where a cylindrical mask is placed over the strut to be analysed. The colour and opacity of the struts and cylindrical mask can be altered by entering RGBA values, in order to optimise the viewing ability of the user.

The cylindrical mask can be created using the **Create cylinder mask** button, or modifying any of the cylinder parameters. These parameters should be varied by the user until the mask is located over the desired strut:

- Cylinder size: r - cylinder radius, and h - cylinder height
- Cylinder location: movement of cylinder in X, Y, and Z directions
- Cylinder rotation: azimuth - rotation of cylinder about its origin, parallel with the z-axis, and elevation - rotation of the cylinder up from the XY plane

Coarse and **Fine** buttons can be used to switch between large and small step sizes, and the size of these steps can be altered using the input boxes.

Examples of masking a strut are shown in Figure 4, with common errors demonstrated. These are inclusion of material from neighbouring struts, and exclusion of protruding strut material. Utilise the rotation and zoom functions of the 3D viewer to ensure that the mask only surrounds the desired strut. If unsure, skipping ahead to the analysis step (use a faster cylinder fit) can reveal if the current mask placement is suitable.

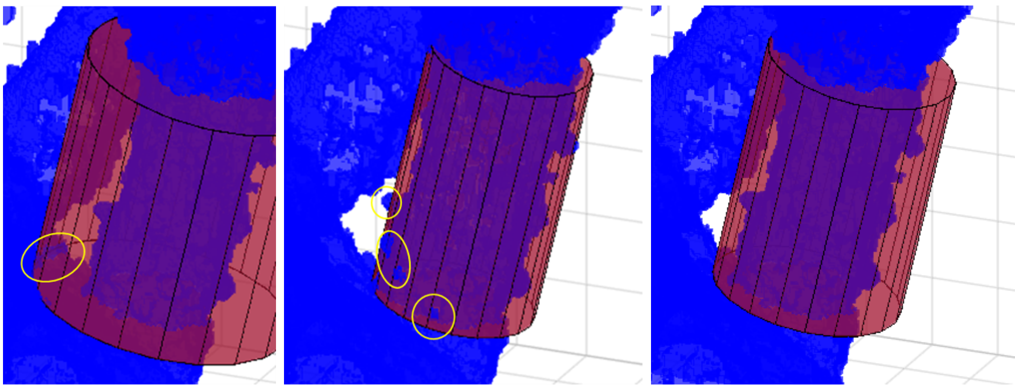

Figure 4: Placement of mask cylinder over strut demonstrating common errors. Left: inclusion of material from neighbouring strut, middle: exclusion of protruding strut material. Right: correct masking of strut.

Once the mask is placed correctly, the **Analyse strut** button can be used to carry out strut analysis. There are currently two options for fitting the strut:

- **Cylinder:** fits a simple cylinder (circular cross-section) to the strut.
- **Elliptic cylinder:** fits an elliptic cylinder (elliptical cross-section). An elliptic cylinder is often needed to properly describe the shape of additively manufactured struts, however this computation is slower due to the evaluation of elliptic integrals.

5 Strut analysis

Results of the strut analysis are shown in the **Analysis** tab. The main visualisations are the strut (left panel), level strut (middle panel), and strut surface (right panel), shown in Fig. 5. The panel on the top right can be used to set the pixel size and switch between visualisations in pixel and µm units, but it should be noted that this only affects the displayed visualisations (and exported versions of these in

.csv and .stl format). The pixel size option does not affect the results table, which is always displayed in pixel units. The visualisations show:

- **Strut:** the portion of the mesh inside the cylinder mask. This can be used to ensure that the mask has been positioned appropriately. The fitted strut axis is shown in red.
- **Level strut:** the strut, rotated so that the strut axis lies along the x-axis. The fitted strut cross-section is shown in red.
- **Strut surface:** extracted (unwrapped) and resampled strut surface, showing deviations from the fitted shape (cylinder or elliptic cylinder). Resampling is carried out to remove overhanging features and convert to an even XY grid format, to ensure accurate calculation of surface roughness parameters.

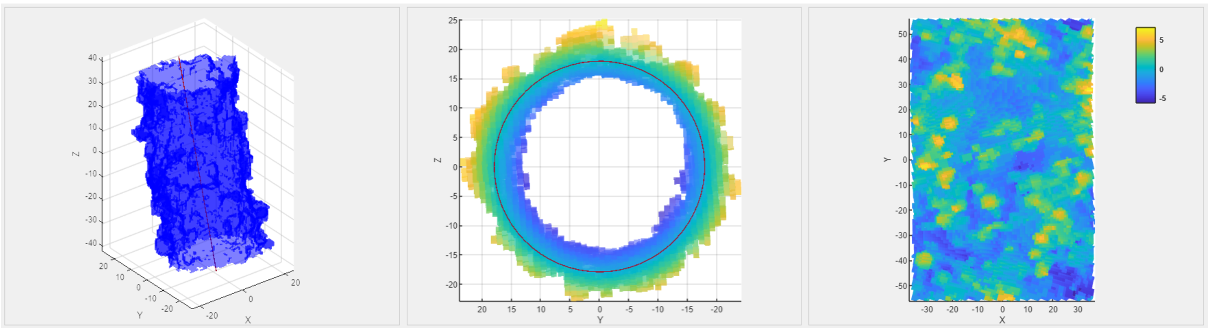

Figure 5: Analysis graphs showing the analysed strut (left), level strut (middle) and strut surface (right).

The control panel on the left (Fig. 6) contains controls for each of the three plots in different tabs.

- The view can be switched between X, Y, and Z axes, and a default 3D perspective. 3D rotation is also possible using the mouse.
- **Reset** returns the selected plot to its default settings, and **New window** opens a new figure window containing the current view.
- The **Opacity** of the strut can be set.
- **Show surface colour on strut** allows the deviation from the fitted shape to be mapped onto the strut as a colourscale, to highlight peaks and valleys.
- The **fitted cross-section** (cylinder or elliptic cylinder) can also be plotted on the level strut.
- Each of these plots can also be exported as an .stl mesh or set of XYZ points in a .csv file.

Results from strut analysis are stored in the table in the analysis tab. This table can be saved using the **Save table** button, which opens a dialog box where the user can navigate and select where to store the results as a .csv file. Once this is set, the saved .csv file can be updated with later results using the **Update saved table** button.

By selecting a row in the table and right-clicking, the context menu allows the user to delete selected rows, or to switch to viewing a previous analysis. The **View** column in the table displays an “x” for the entry that is currently being displayed in the plots above.

The entries in the table are defined below. Except for angles, all values are in pixel units:

- **View:** displays an “x” for the analysis currently being displayed. Empty otherwise.

Import and processing settings:

- **Slice dataset:** common file name prefix for the image slices imported.
- **cb:** Crop buffer - buffer pixels value set during import and cropping.

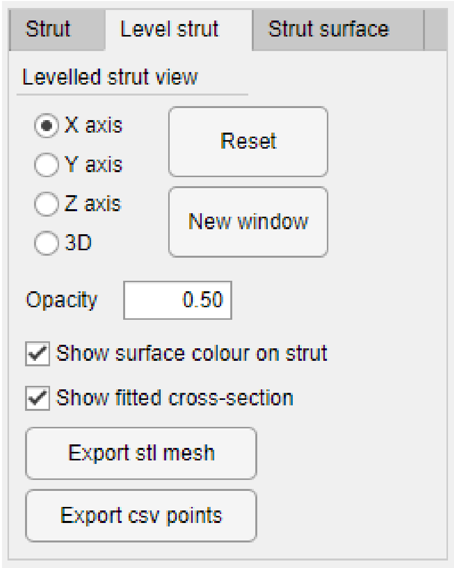

Figure 6: Control panel for altering visualisation options in analysis tab.

- **rs:** Region size - sample region size set during generate of the sub-region mesh.
- **X, Y, Z:** Co-ordinates of the centre of the generated sub-region. Can be manually added in sample region tab to recreate analysis.
- **VxF:** Voxel filter: minimum object size, disconnected objects smaller than this removed during sample sub-region generation. If removal not carried out, VxF = NaN.
- **C...:** Cylinder mask parameters, can be manually added in mask tab to recreate analysis.
  - **Cr:** Cylinder radius
  - **Ch:** Cylinder height
  - **Ctx:** Cylinder translation in X direction
  - **Cty:** Cylinder translation in Y direction
  - **Ctz:** Cylinder translation in Z direction
  - **Caz°:** Cylinder rotation azimuth angle $^\circ$
  - **Cel°:** Cylinder rotation elevation angle $^\circ$

Calculated results:

- **cd:** Centre distance - radial distance from centre of imported volume to strut centroid.
- **El°:** Elevation angle of strut above XY plane.
- **D1:** Primary diameter - diameter of cylinder fitted to strut.
- **D2:** Secondary diameter - second diameter of cylinder fitted to strut. Only used when elliptic cylinder is fitted. If simple cylinder is fitted, D2 = NaN.
- **Dfmx:** Maximum Feret diameter of the strut perpendicular to its axis.
- **Dfmn:** Minimum Feret diameter of the strut perpendicular to its axis.
- **Sa:** Areal surface roughness Sa - arithmetic mean surface roughness.
- **Sq:** Areal surface roughness Sq - root mean square surface roughness.
- **Sz:** Areal surface roughness Sz - maximum peak to valley height.

Other surface roughness and texture parameters can be calculated from the raw surface data, which can be exported to .csv using the control panel for the strut surface plot.

> **Important:** When interpreting calculations of the surface roughness, the pixel size should be carefully considered. This should be small enough to easily resolve the main features that contribute to the surface roughness.

References

[1] Qianqian Fang and David A. Boas. Tetrahedral mesh generation from volumetric binary and grayscale images. In *Proceedings - 2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro, ISBI 2009*, pages 1142–1145, 2009.

[2] iso2mesh: a Matlab/Octave-based mesh generator. http://iso2mesh.sourceforge.net/cgi-bin/index.cgi?Home.