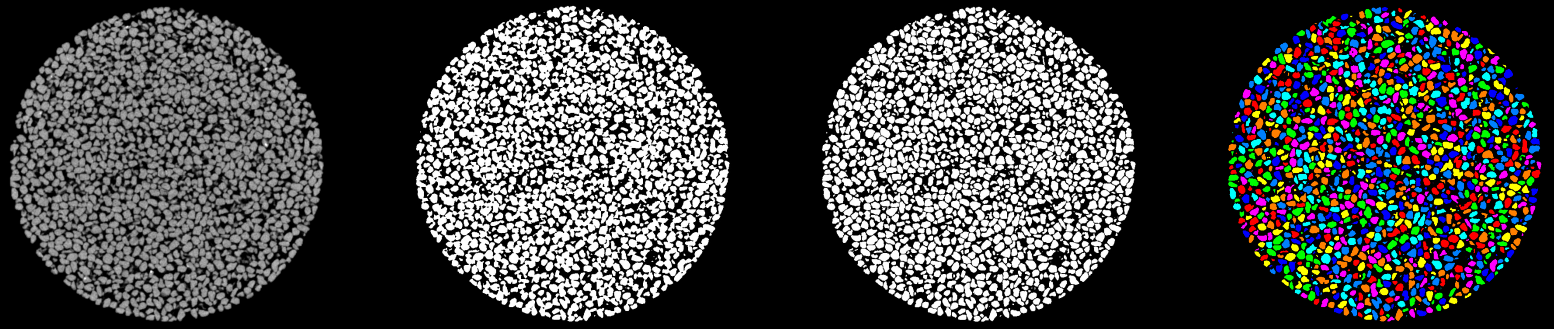
SWEDNESS/LINXS DOCTORAL
COURSE ON NEUTRON IMAGING

3D IMAGE ANALYSIS TUTORIAL

Stephen Hall

Division of Solid Mechanics, Lund University, Sweden
& Lund Institute for Advanced Neutron and X-ray Science (LINXS)



SwedNess

LUND INSTITUTE OF ADVANCED NEUTRON AND X-RAY SCIENCE

Software

- Matlab – you should have the Matlab® available from the university, which includes the image processing toolbox (this toolbox is necessary for the exercise).

- Imagej/Fiji: runs in Java and can be downloaded for free and with little installation from :

  - http://fiji.sc/Fiji

Objectives

- The objective of this exercise is to learn about how information about material structure that can be important for understanding mechanics/creating mechanics models can be extracted from images

- In this case we will be looking at 3D images from x-ray tomography

- Three key aspects will be explored, data visualisation, image segmentation and object quantification.

Software

- Many different software tools exist for image analysis and visualisation.

- Here we will use

  - **Matlab®**, as it has a powerful image processing toolbox and because we all have access to it through the university (Matlab is not so good for 3D visualisation though)

  - **Imagej/Fiji**, which is free (originally from the US National Institute for Health) and has a lot of nice tools for image analysis, processing and visualisation (we will only e using the visualisation part here)

Software

- Matlab – you should have the Matlab® available from the university, which includes the image processing toolbox (this toolbox is necessary for the exercise).

- Imagej/Fiji: runs in Java and can be downloaded for free and with little installation from :

  - http://fiji.sc/Fiji

- Could also try these (works on most platforms and has “hardware acceleration for rendering):

- TomViz: https://tomviz.org

- Vaa3D: www.vaa3d.org

- Drishti: https://sf.anu.edu.au/Vizlab/drishti/

Data sets:

- 3D image of a lego fireman
  - https://www.dropbox.com/sh/xdr2193ubd3fnf6/AADSfqxEHNNlyeJvlTgSQrQsa?dl=0
- 3D image of a foam
  - https://www.dropbox.com/sh/uqaygot5zrvx5bq/AAATjhNJYU3u6YH5GZThWJ_7a?dl=0

Data sets:

- 3D image of a lego fireman
  - 80 kV tube voltage & 7 Watt tube power
  - 0.4x optical magnification
  - Camera binning 4x4
  - Exposure time:
  - 1601 projections
  - Pixel size: 97.66 microns
  - Tomography images rebinned to half size (pixel size: 195.32 microns)
- 3D image of a foam
  - 40 kV tube voltage & 3 Watt tube power
  - 4x optical magnification
  - Camera binning 2x2
  - Exposure time: 5 s
  - 2001 projections
  - Pixel size: 3 microns
  - Tomography images rebinned to half size (pixel size: 6 microns)

- Acquired at the 4D Imaging lab in Lund

Documenting:

Try to make annotated notes on the described image analysis keeping track of the key steps in the processing, including:

- 3D visualisations of the two datasets
- Histograms of the grey-scale values
- Binarisations (including justifications of the values used)
- Labelled volume and described statistical analysis of the foam example

For each processing step it is important to think about and describe what is being done and why (so that if you want to do it again you can!)

Image analysis…

… what is the first step?

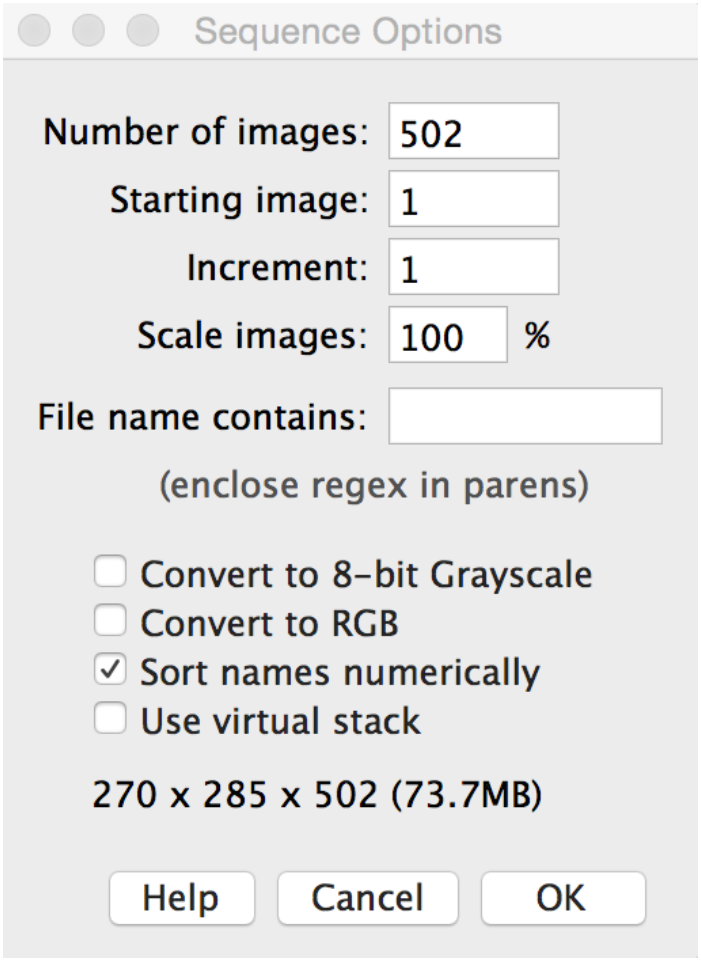
Image analysis…

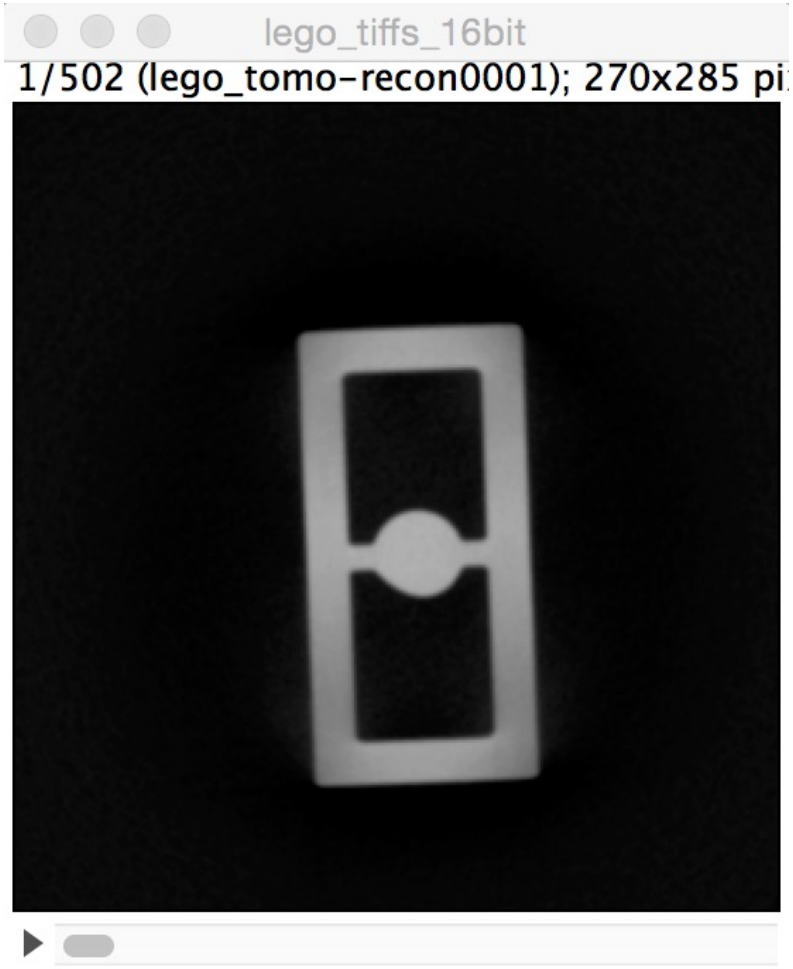
… what is the first step?

… exploration / visualisation of your data

3D visualisation in Fiji

- File > import > image sequence
  - Select “lego_tiffs_16bit” directory

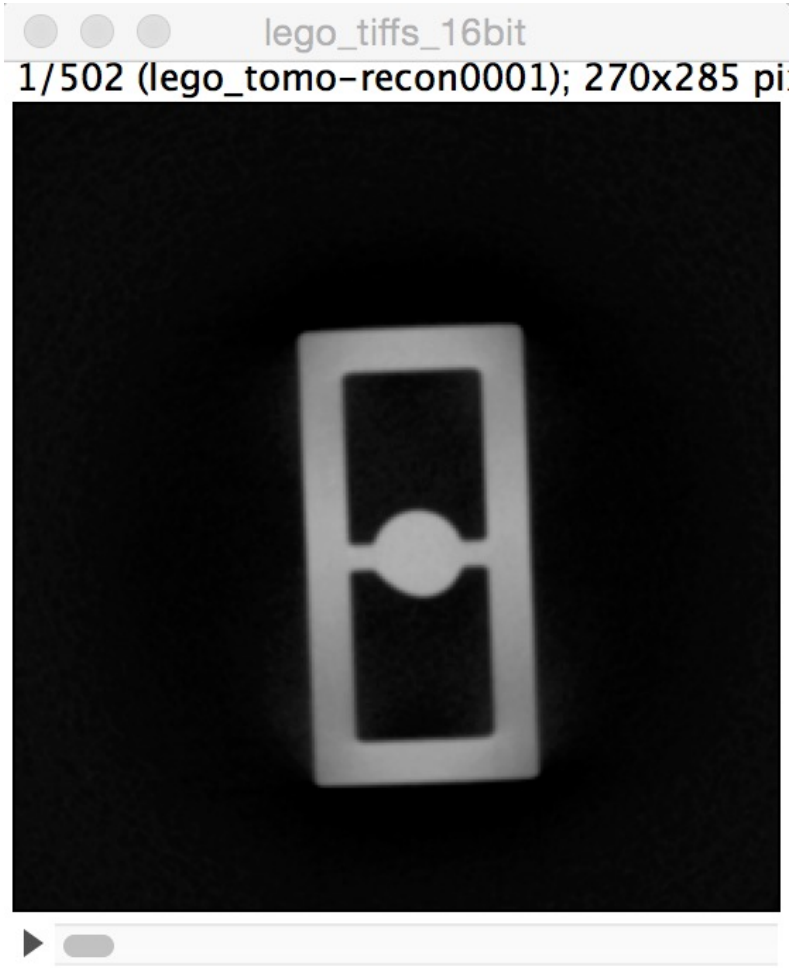




Ignore the error messages in the console (these relate to missing information in the tiff header)

3D visualisation in Fiji

- You now have a “stack” of images, each image corresponds to a reconstructed slice in the tomography volume and you can scroll through the images

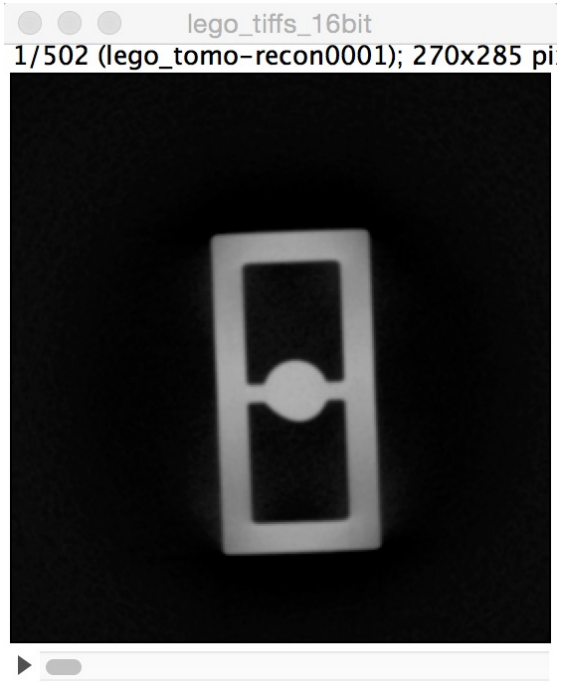


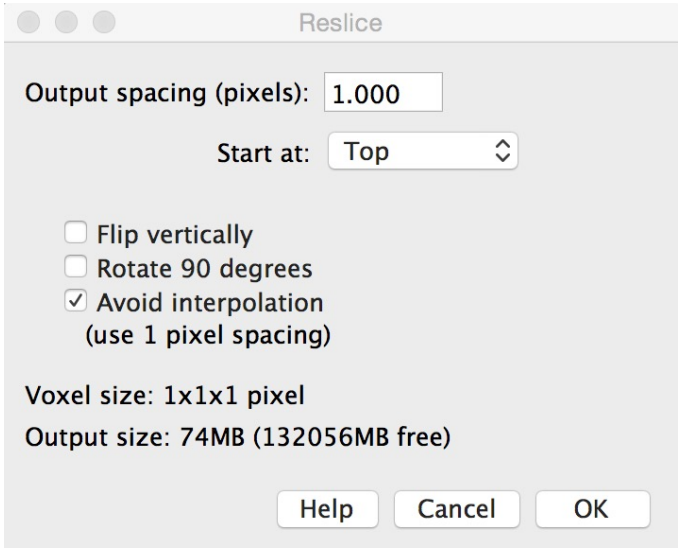
You can change in the colour scale/range using:

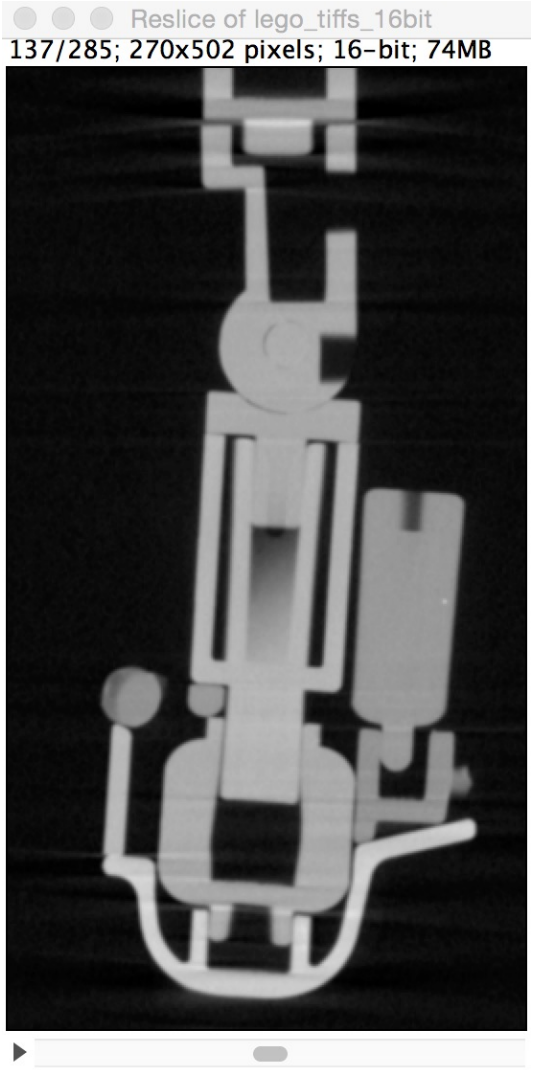
“image > adjust > brightness/contrast”

3D visualisation in Fiji

- You can also “reslice” the images in any way you want (remember, it is just a 3D matrix of intensity values)
- Try “image > stacks > reslice > “from top”

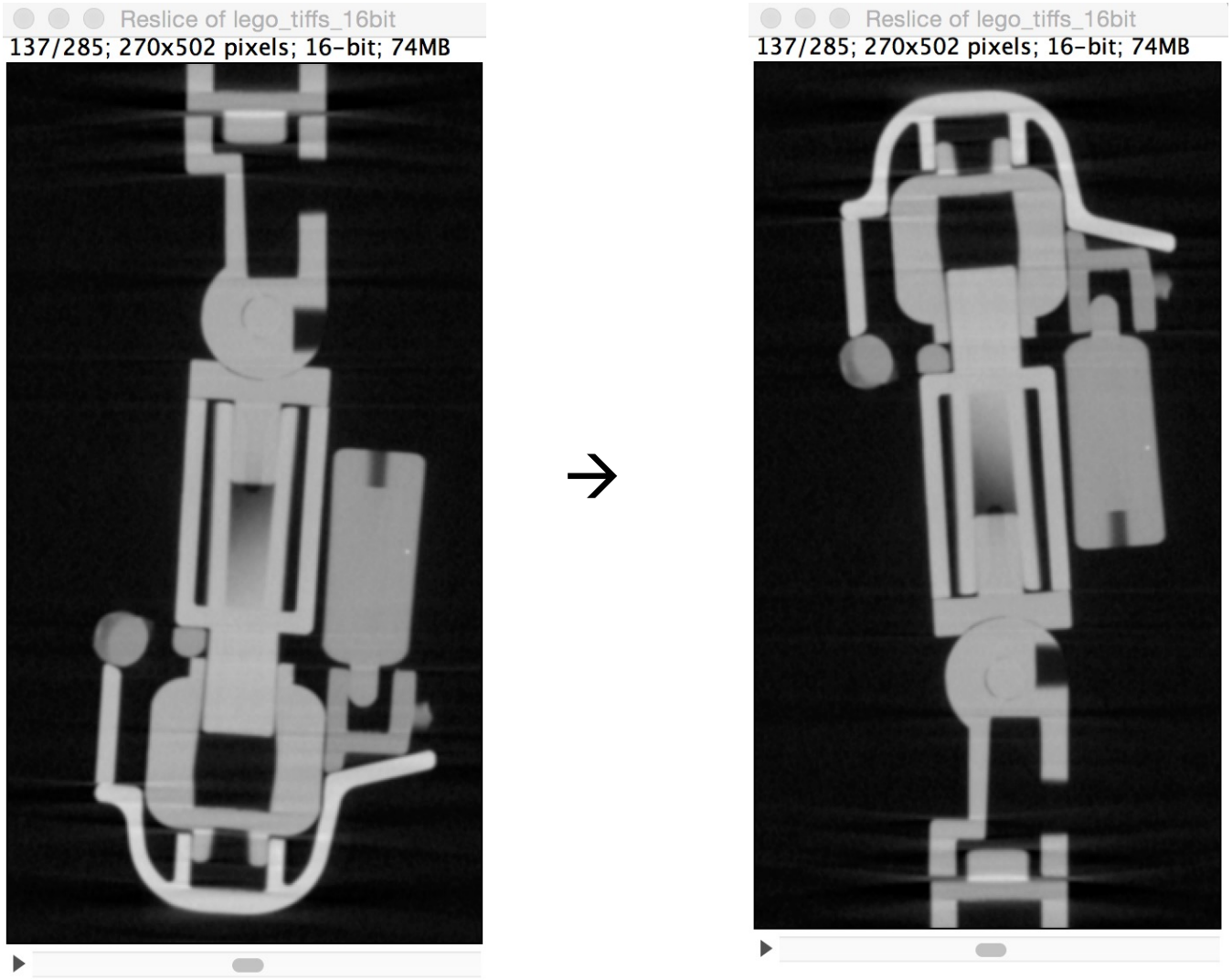






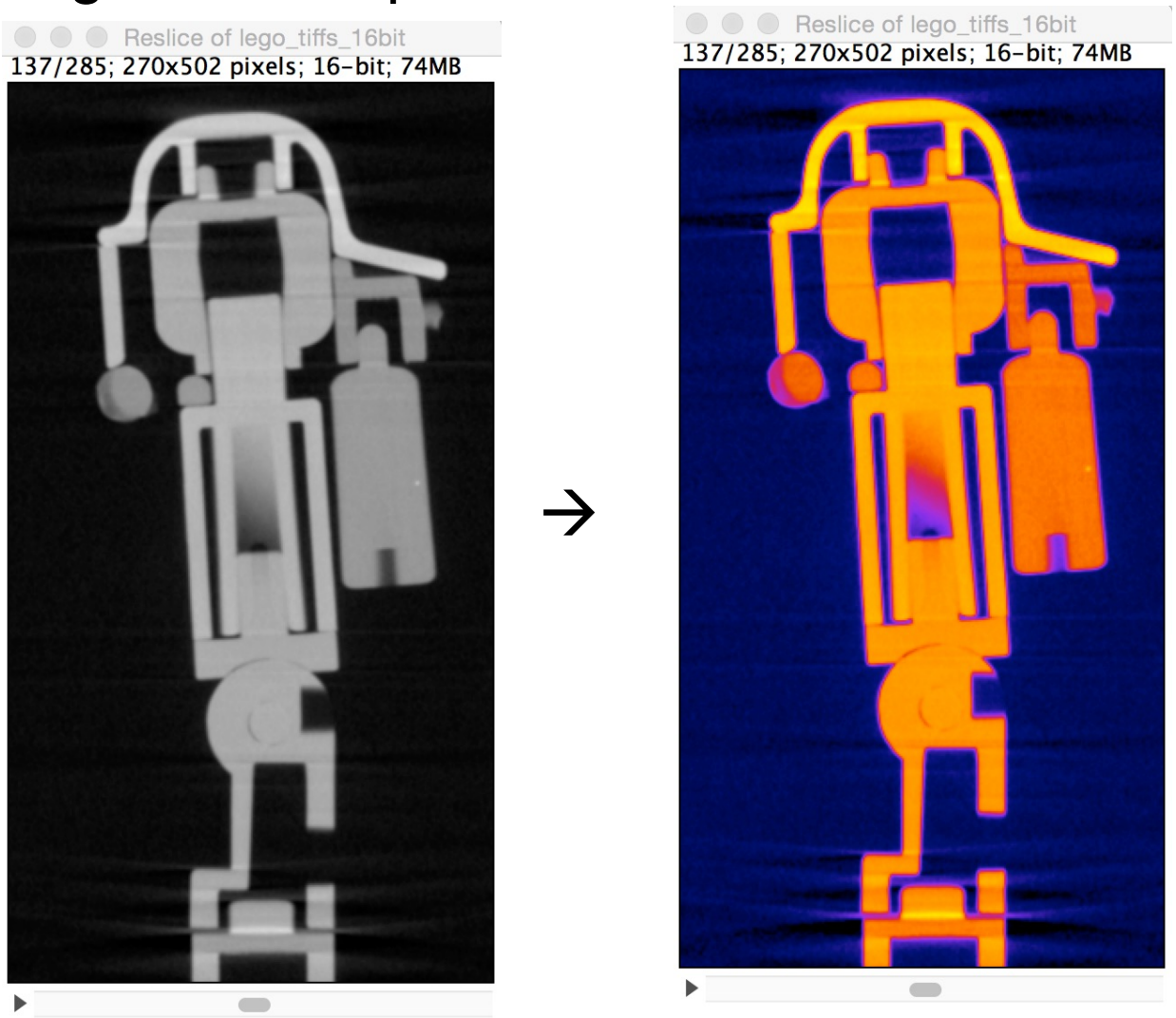
3D visualisation in Fiji

- Flip the image, if you want:
  - Image → transform → flip vertically



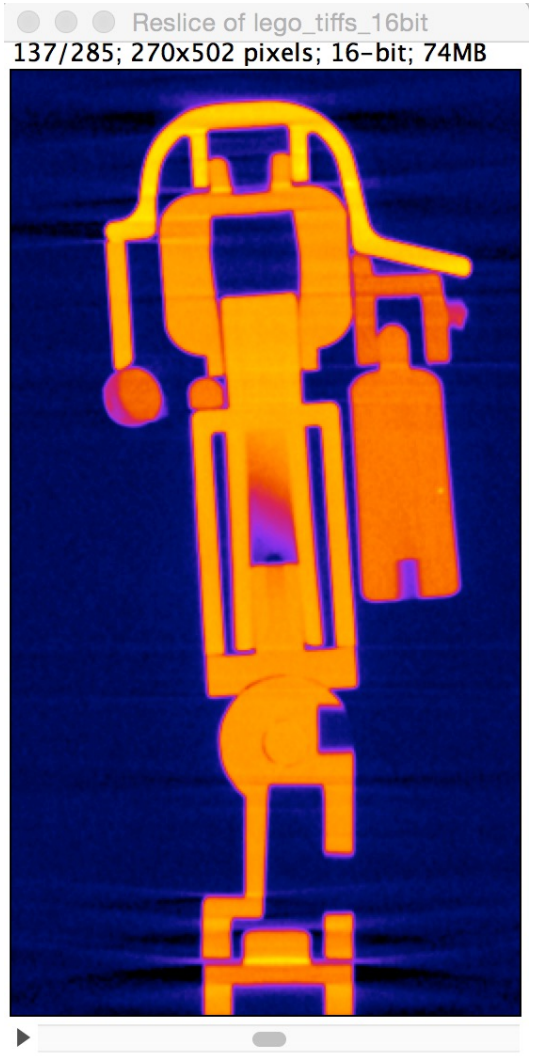
3D visualisation in Fiji

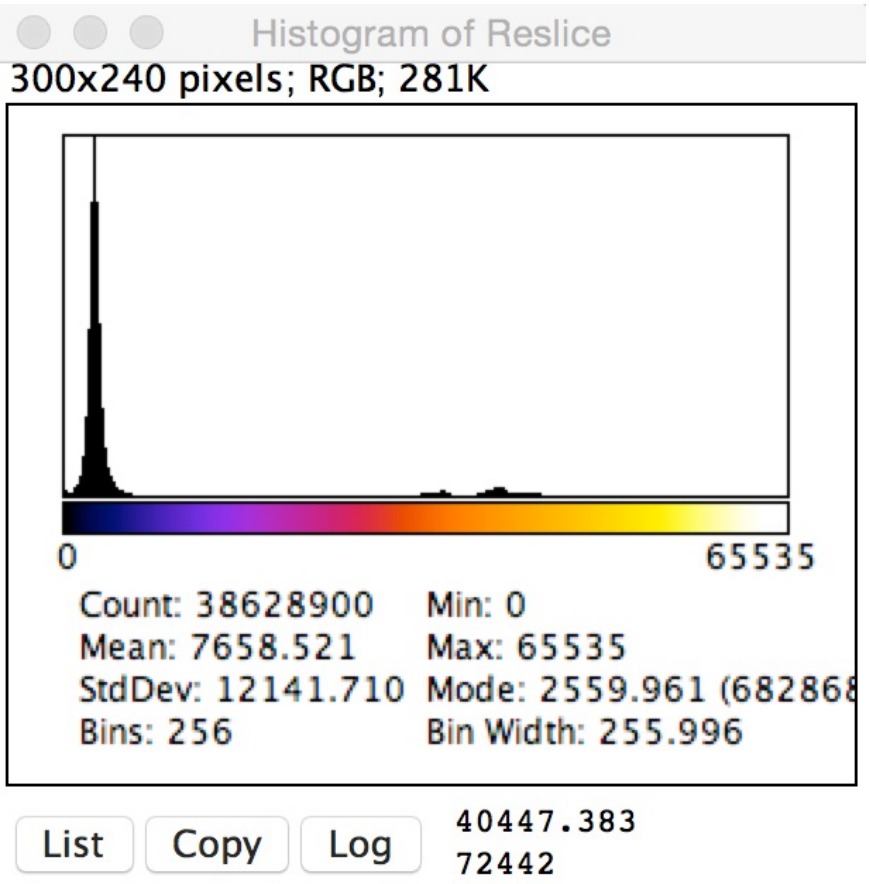
- Choose a new “colourmap”, if you want:
  - Image > look up tables > fire



3D visualisation in Fiji

- Look at the “histogram” of intensities
  - Analyze > histogram (include all images)

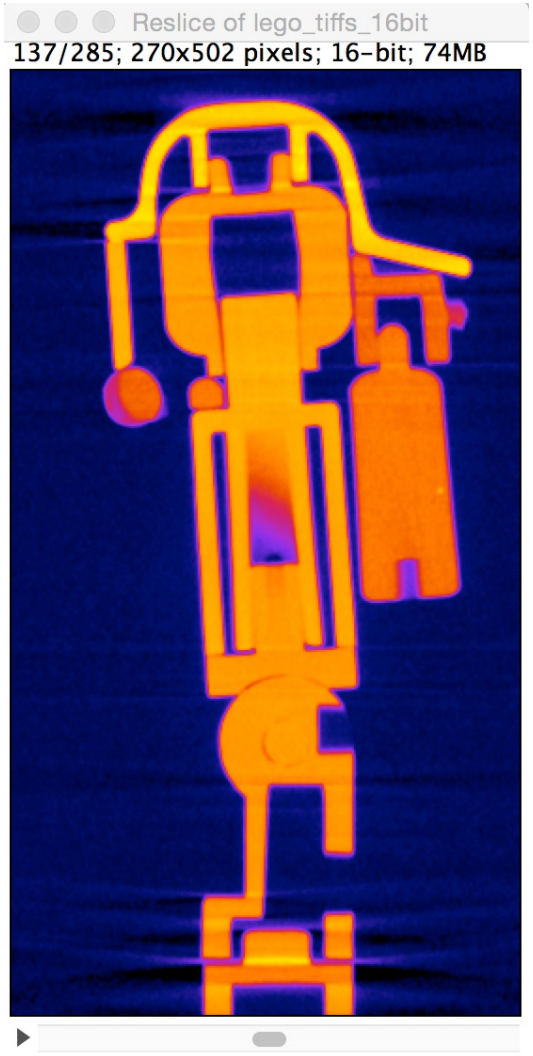


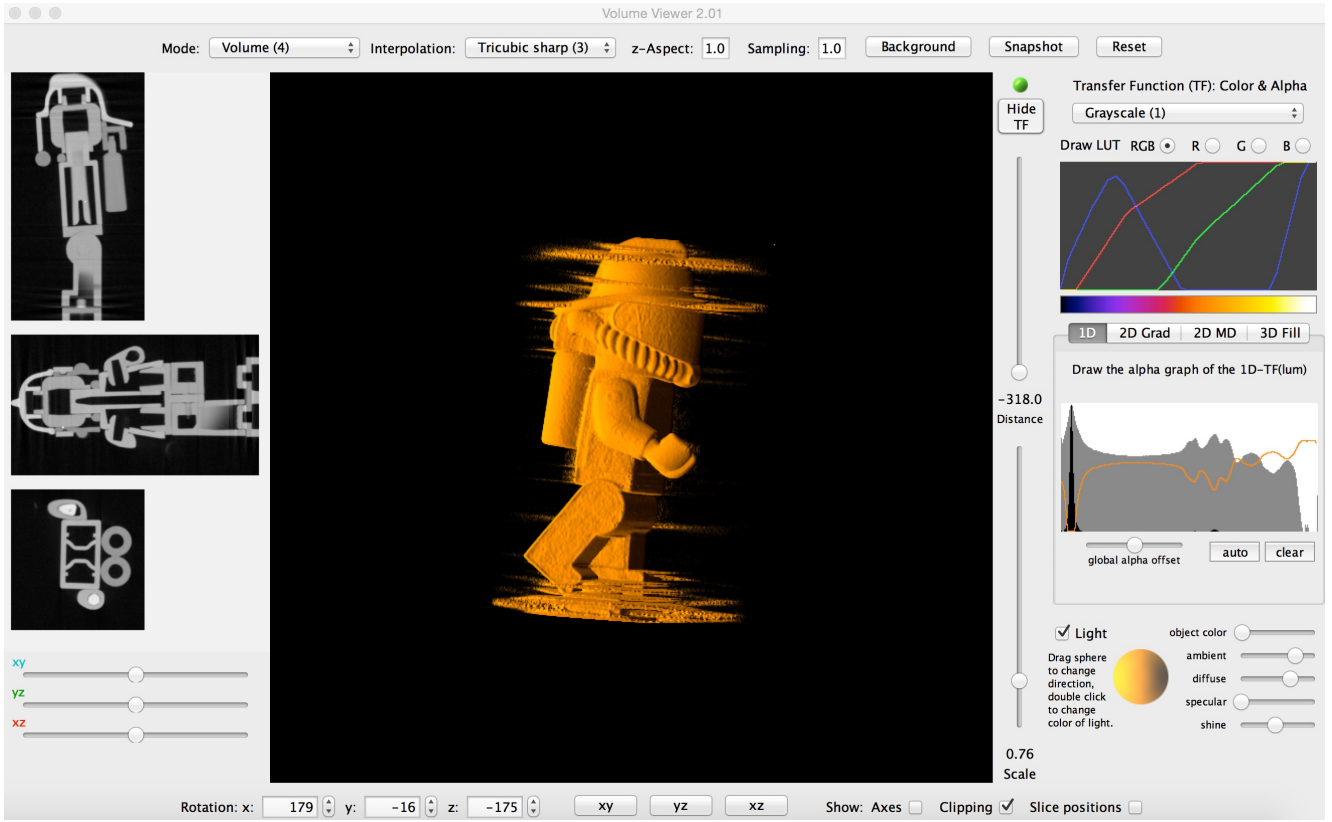


Try with a log scale

3D visualisation in Fiji

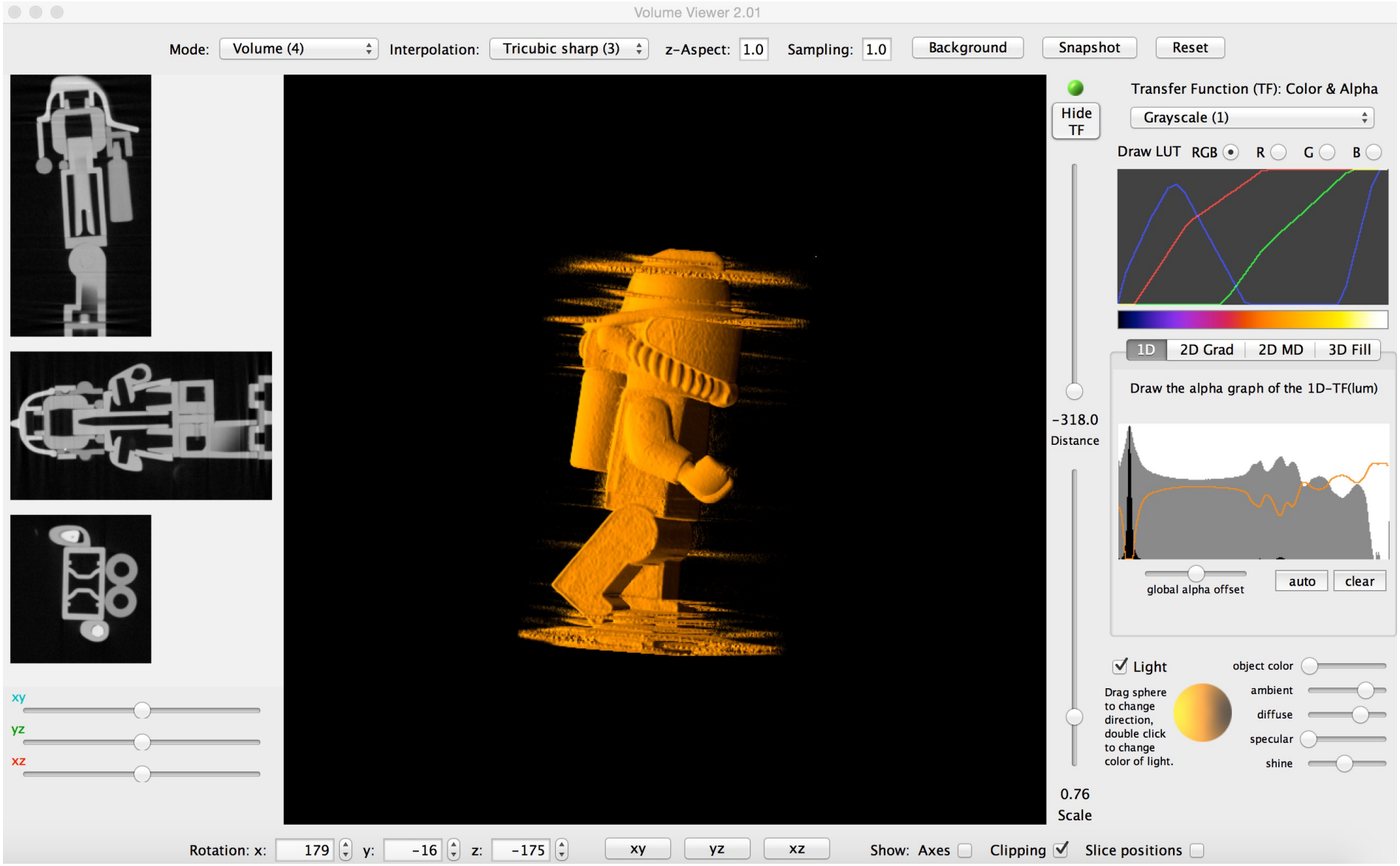
- Now let's look at it in 3D:
  - Plugins > Volume Viewer





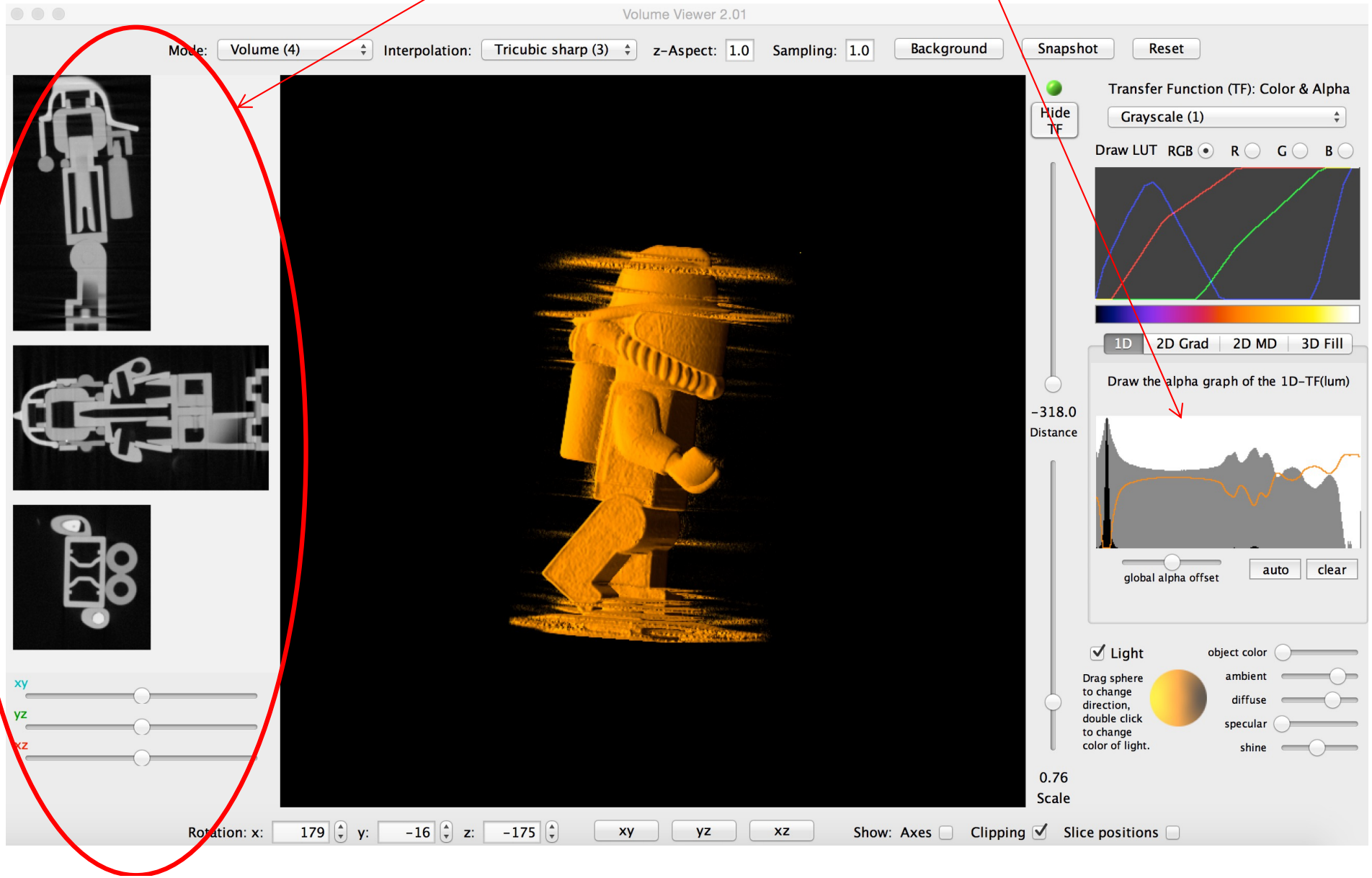
3D visualisation in Fiji

…and play!



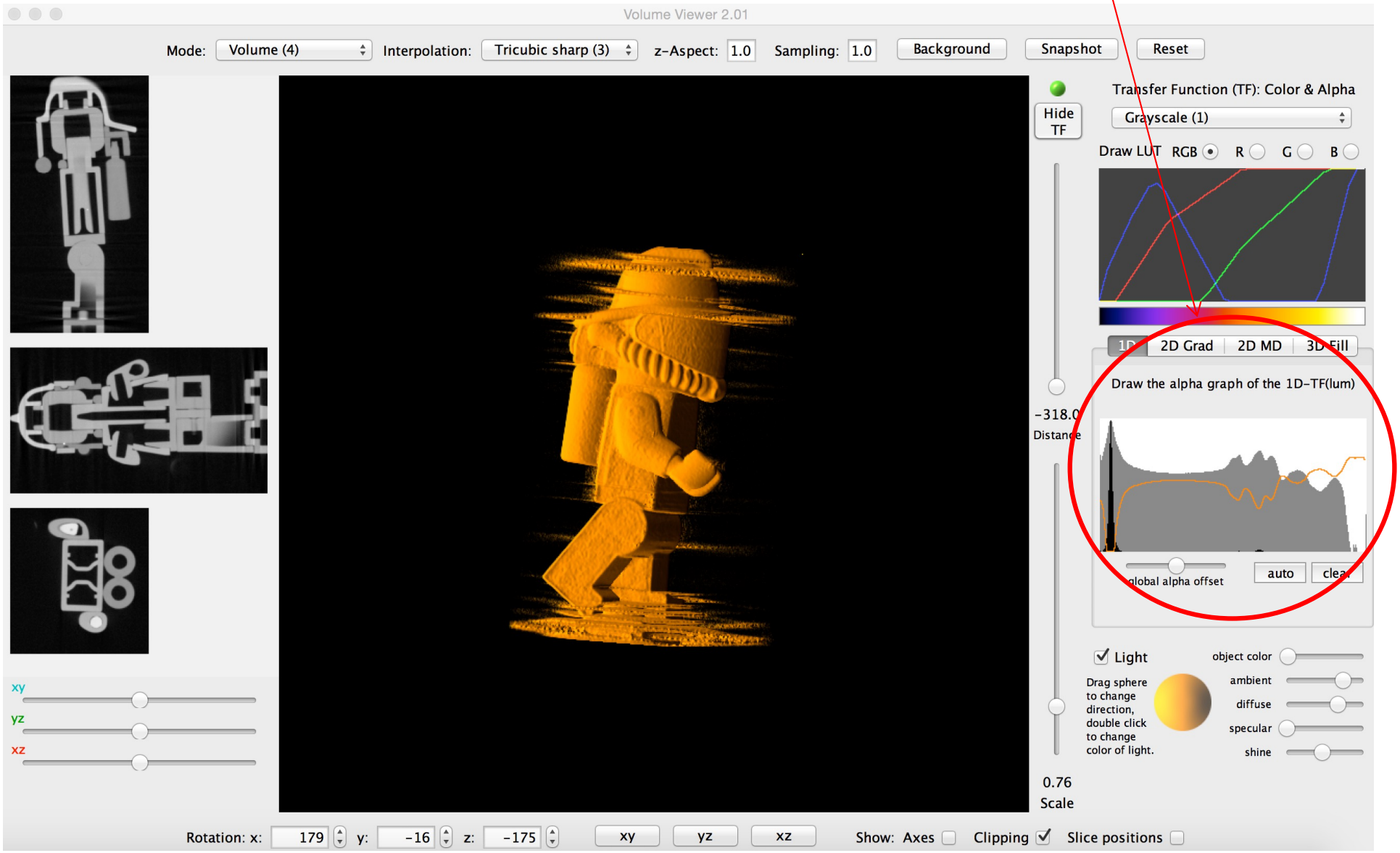
3D visualisation in Fiji

Orthogonal slices – move mouse over slices and you'll see where the grey-scale lies in the histogram by the blue line



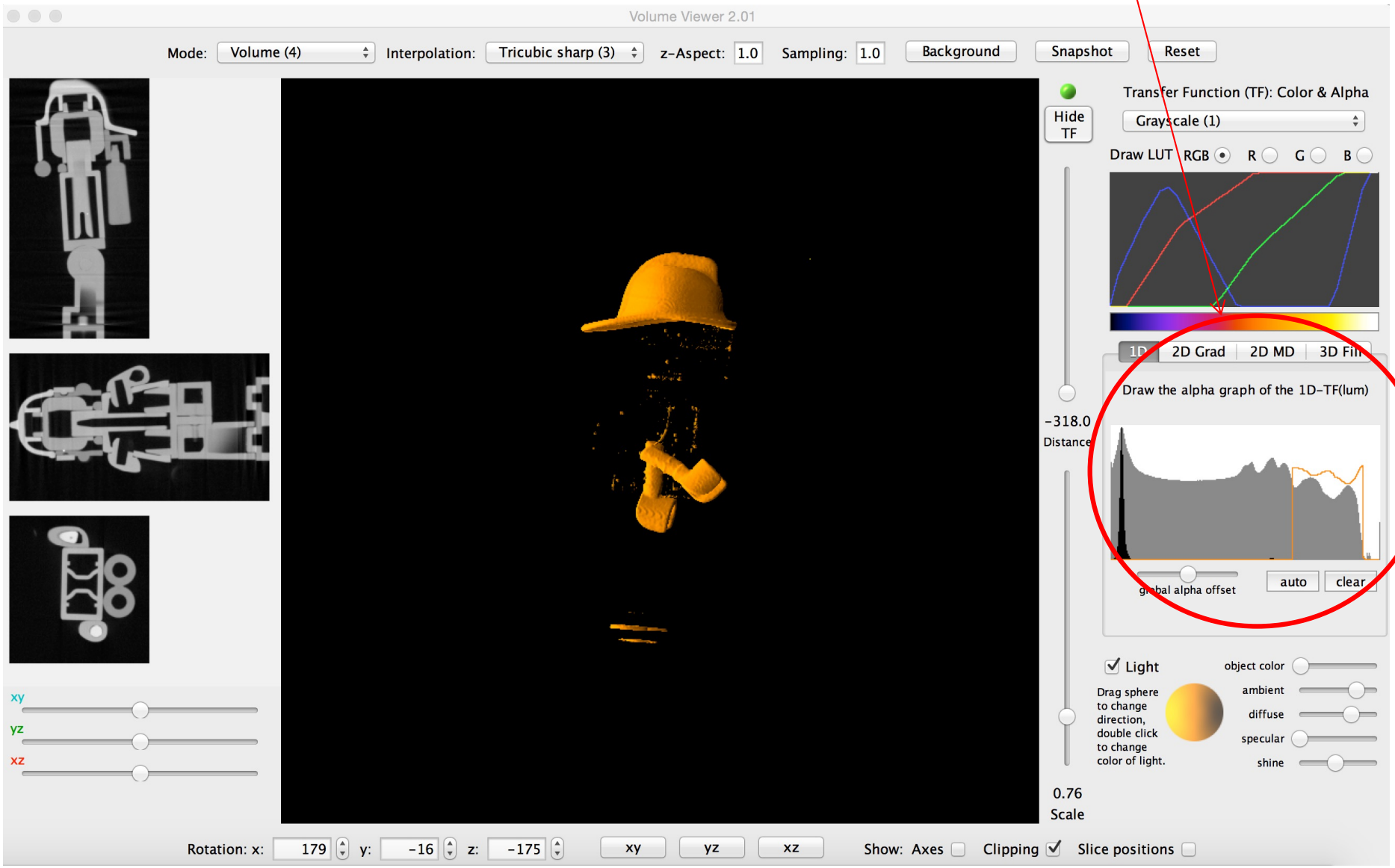
3D visualisation in Fiji

Histogram: indicates grey-scale values in image. Orange line is the “alpha channel (transparency) – use the mouse to change this profile and “turn on/off” parts of the image

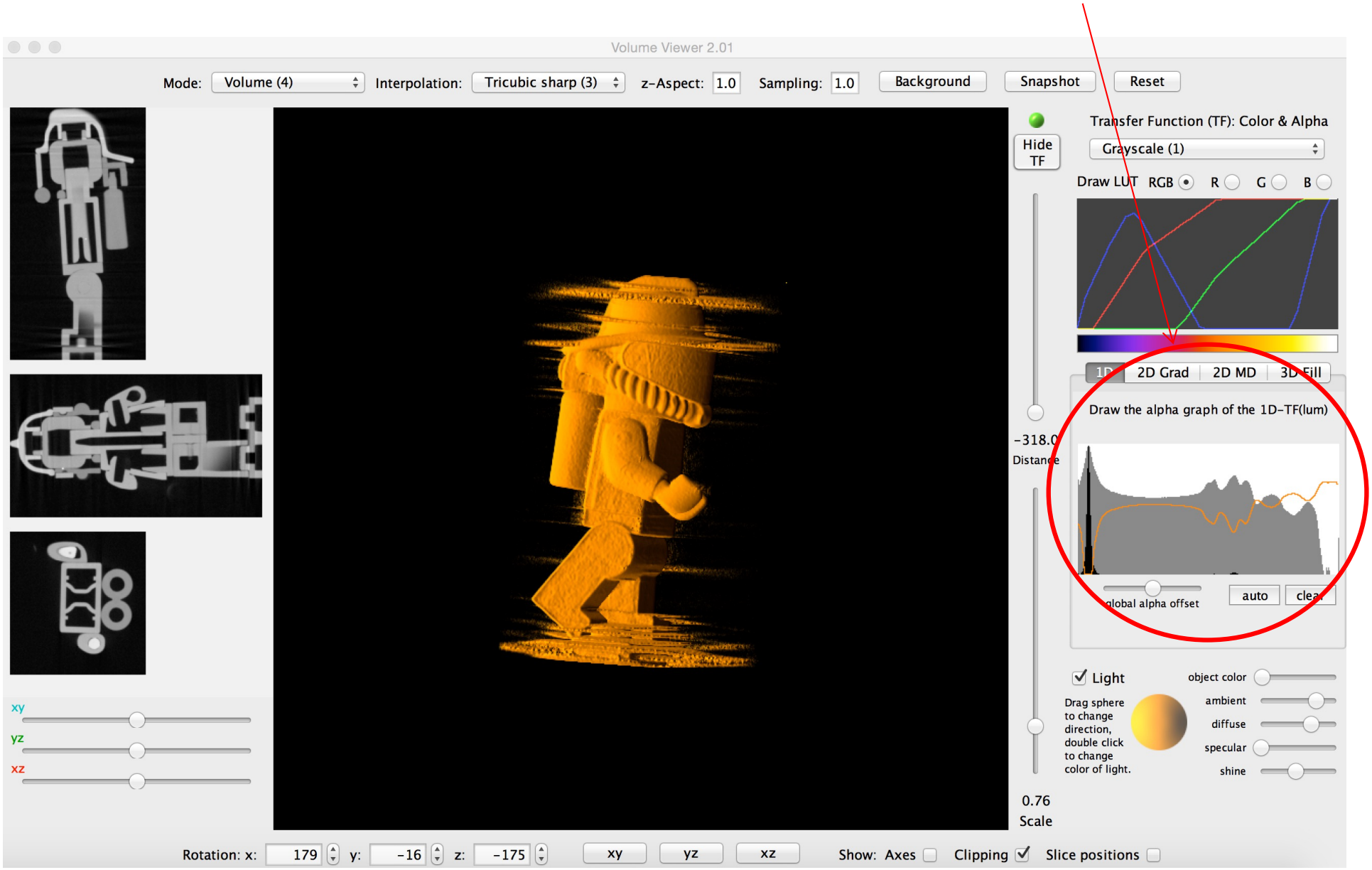


3D visualisation in Fiji

Histogram: indicates grey-scale values in image. Orange line is the “alpha channel (transparency) – use the mouse to change this profile and “turn on/off” parts of the image



3D visualisation in Fiji



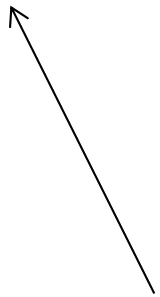
Binarisation

- Possible in Fiji, but we're going to do it in Matlab

Binarisation

- Read in and view data

```
clear
close all

imstart = 1;
imend = 500;

for i = imstart:imend
 data(:,:,i-imstart+1)=im2double(imread(...
    strcat('./lego_tiffs_16bit/lego_tomo-recon',num2str(i,'%.4d'),'.tif')));
end
```

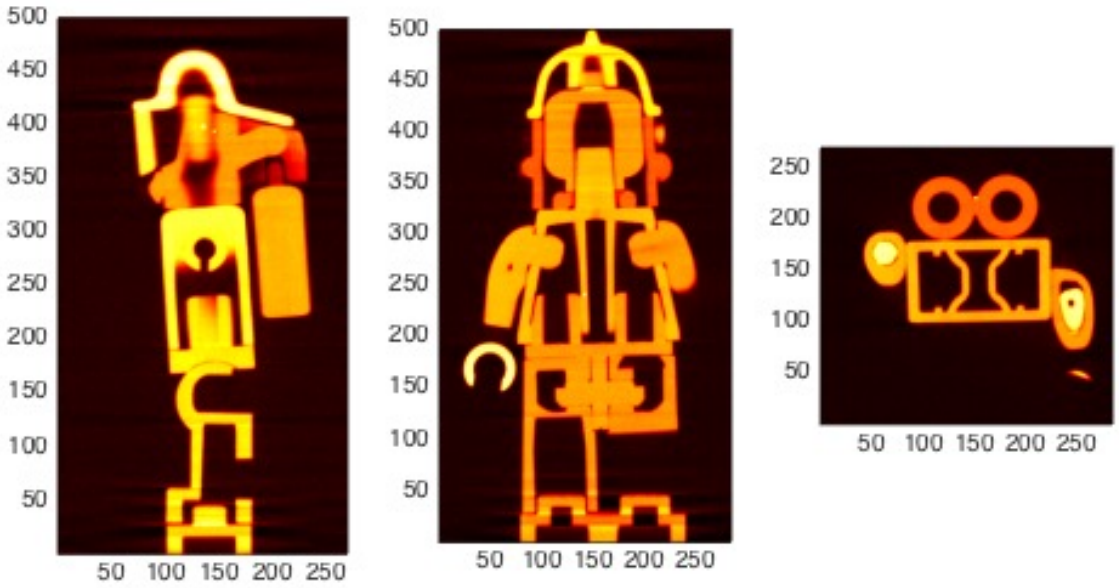


Might need to change the diectory

Binarisation

- Read in and view data

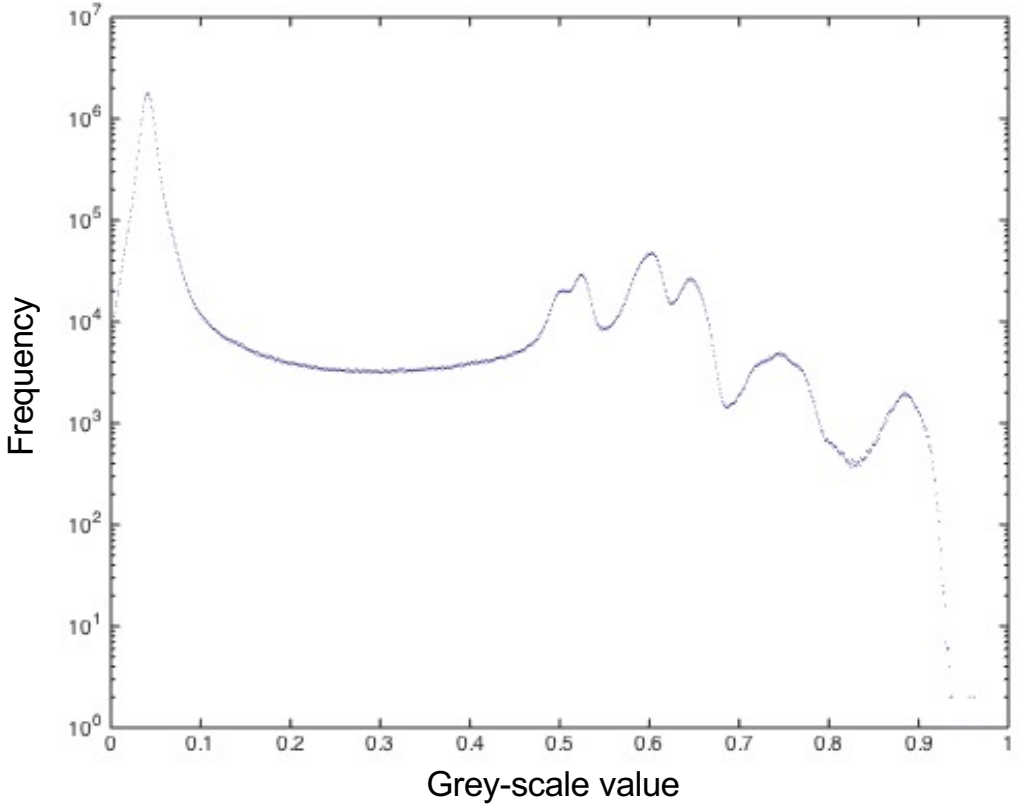
```
% extract and plot slices through volume

slice1(:,:) = data(100,:,:);
slice2(:,:) = data(:,150,:);
slice3(:,:) = data(:,:,250);

figure(1)
subplot(1,3,1)
pcolor(slice1')
shading flat
daspect([1 1 1])
subplot(1,3,2)
pcolor(slice2')
shading flat
daspect([1 1 1])
subplot(1,3,3)
pcolor(slice3')
shading flat
daspect([1 1 1])

% change colormap (if you want to!)
colormap(hot)
```



Binarisation

- Histogram

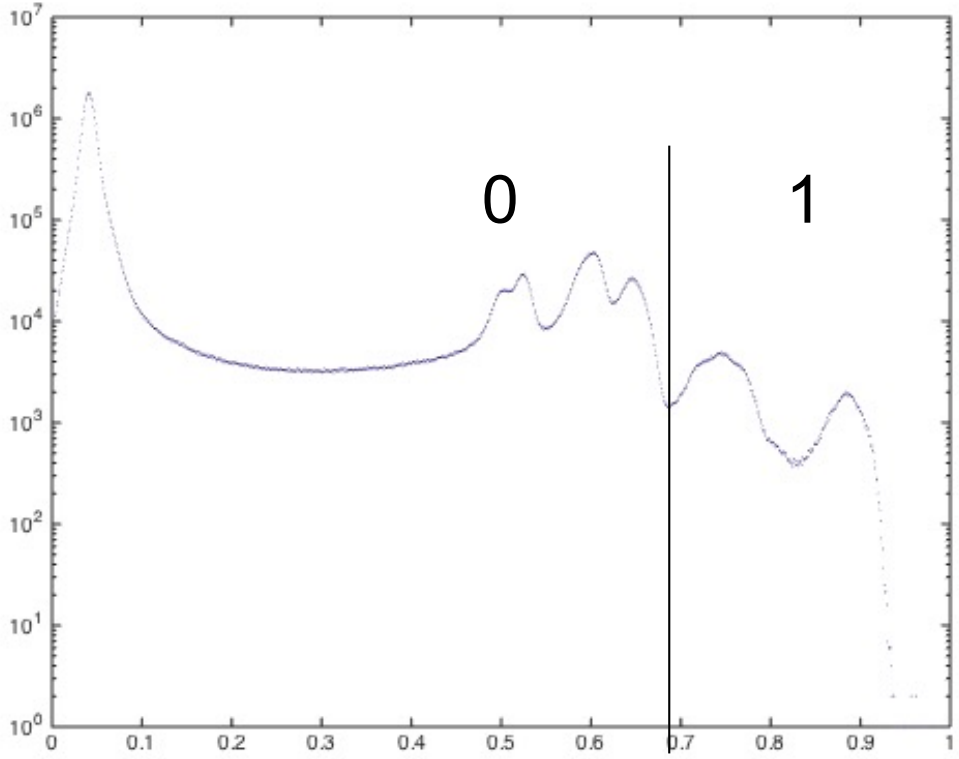
```
% make histogram of grey-scale values to see different intensities
% plot histogram
figure(2)
histogram(data,1000)
% define yscale as logarithmic to see
%smaller peaks in histogram
set(gca,'YScale','log')
```



How many material “phases” do we have?

Binarisation

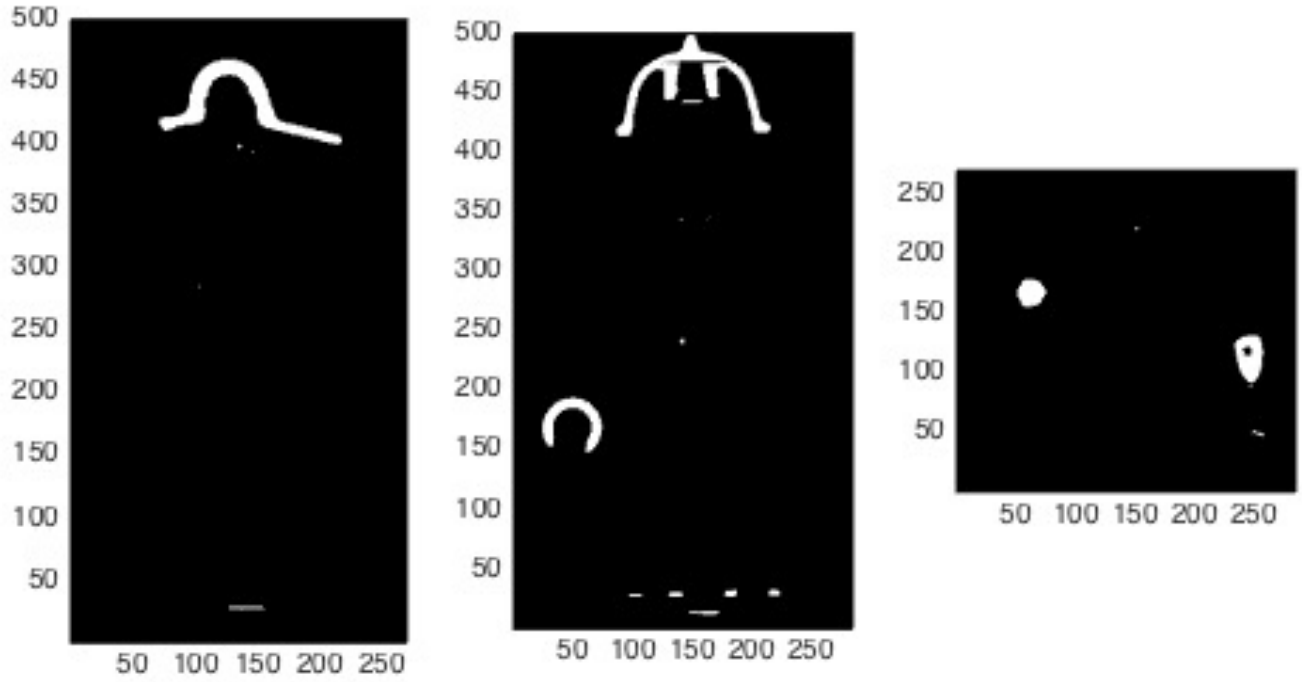
- Thresholding
  - Select threshold “level”




```
threshold = 0.686;

%binarise

bin = data;
bin(bin<threshold) = 0;
bin(bin>threshold) = 1;
```

Segmentation

- Plot slices through binary volume, as before

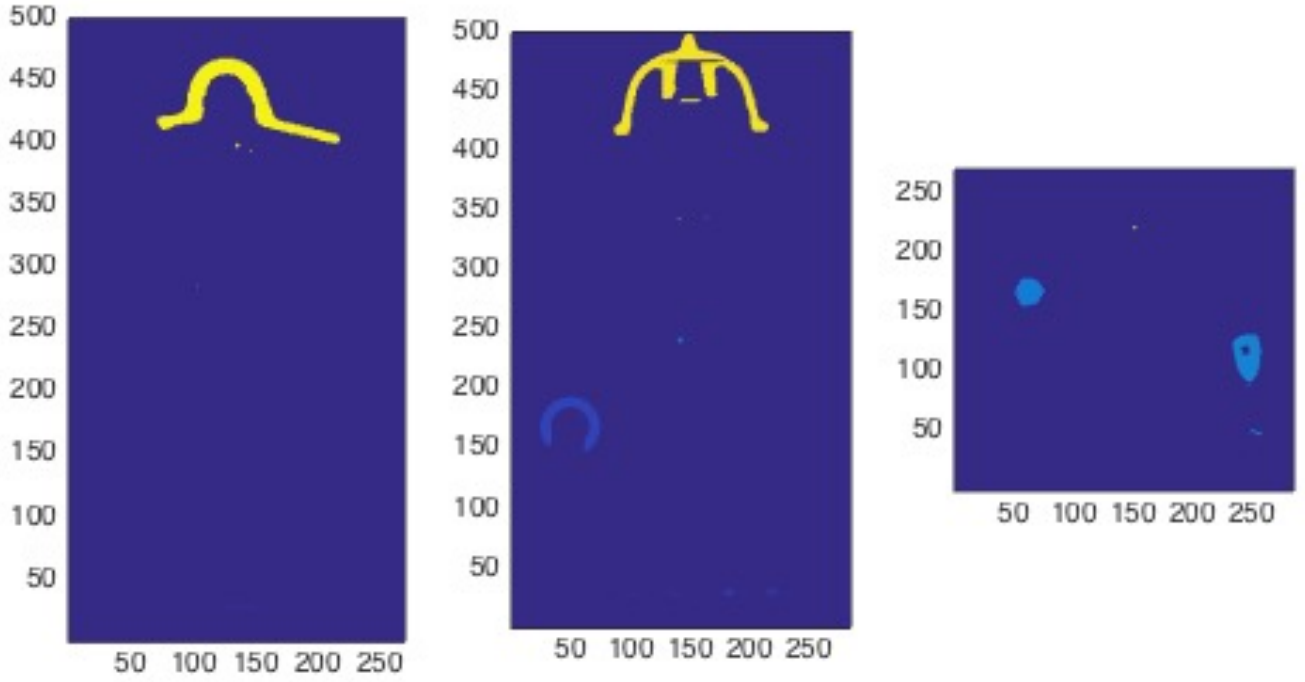


- And repeat for other “phases”, as you wish
- You could even assign a different value to each “phase”…

Segmentation

- Connected components – group together and label connected “objects”

```
CC = bwconncomp(bin);

L = labelmatrix(CC);

slice1(:,:) = L(100,:,:);
slice2(:,:) = L(:,150,:);
slice3(:,:) = L(:,:,250);

…
```

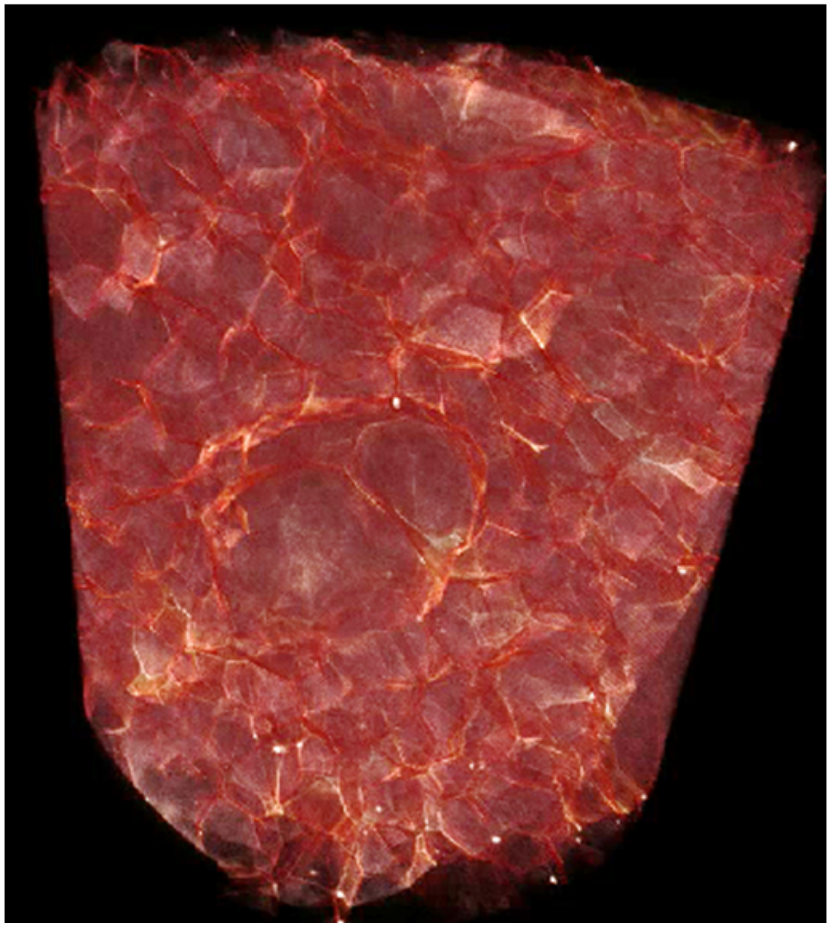


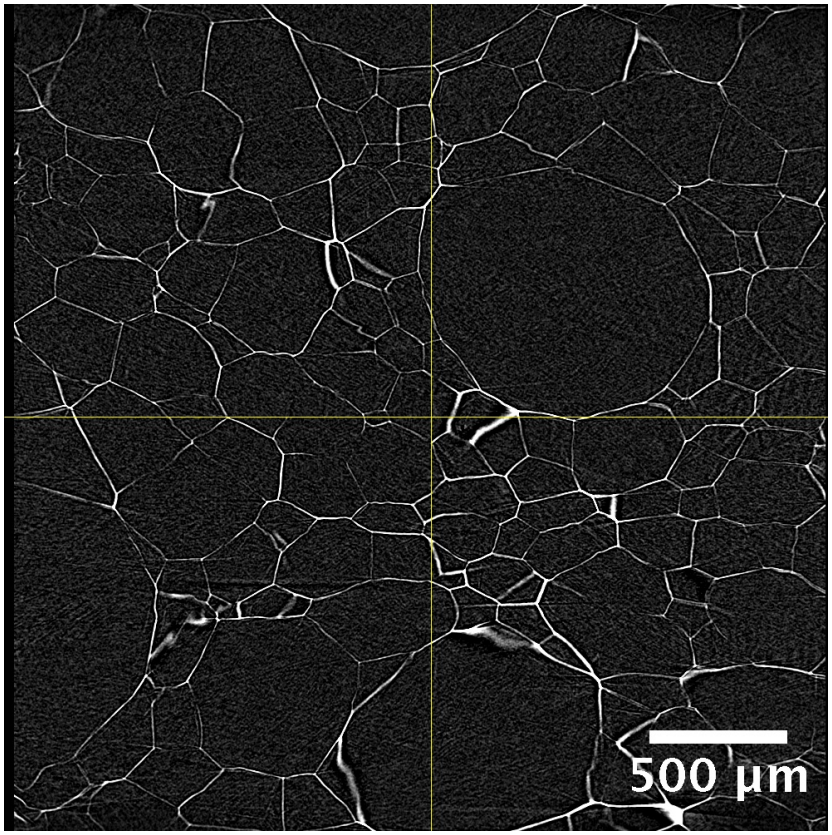
Try visualising in Volume Viewer – now you can isolate the hat/hands, for example

- Some notes for quantification :

- Volume of components:
  - % count number of pixels in each label (volume):
  - stats = regionprops3(L,’Volume');
  - volume = cat(1, stats.Volume);
  - Plot as histogram (hist(volume))
  - Identify which label corresponds to the hat, for example

OK, so now for a more “real”/”complex” example…





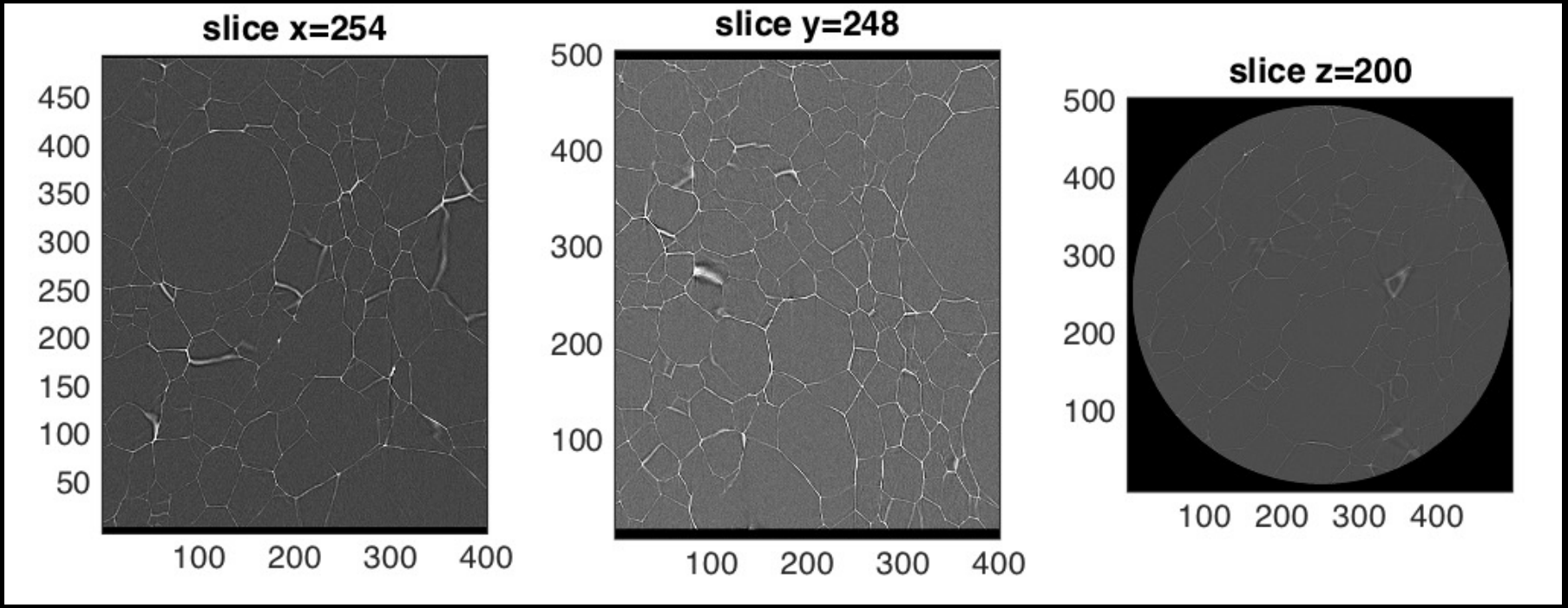
Images in: foam2_40kV_3micron_16bit

Structural imaging and characterisation: 3D image analysis, e.g.,

How do we see/identify and characterise microstructure?

Cellulose-based foam

Voxel width of 3 μm
Field of view 3 mm



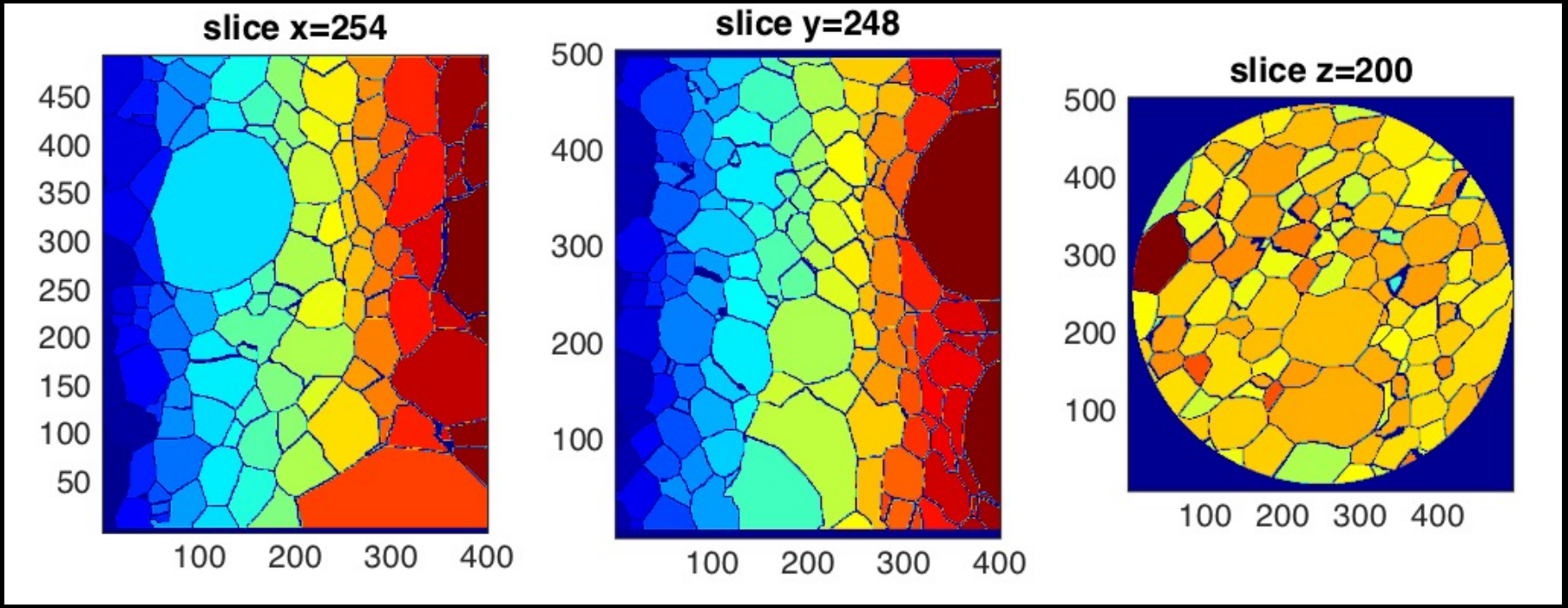
Collab. L. Bergström, S
Gordeyeva et al., 2016,

Structural imaging and characterisation: 3D image analysis, e.g.,

How do we see/identify and characterise microstructure?

Cellulose-based foam

Voxel width of 3 µm
Field of view 3 mm



Watershed segmentation (in Matlab®)

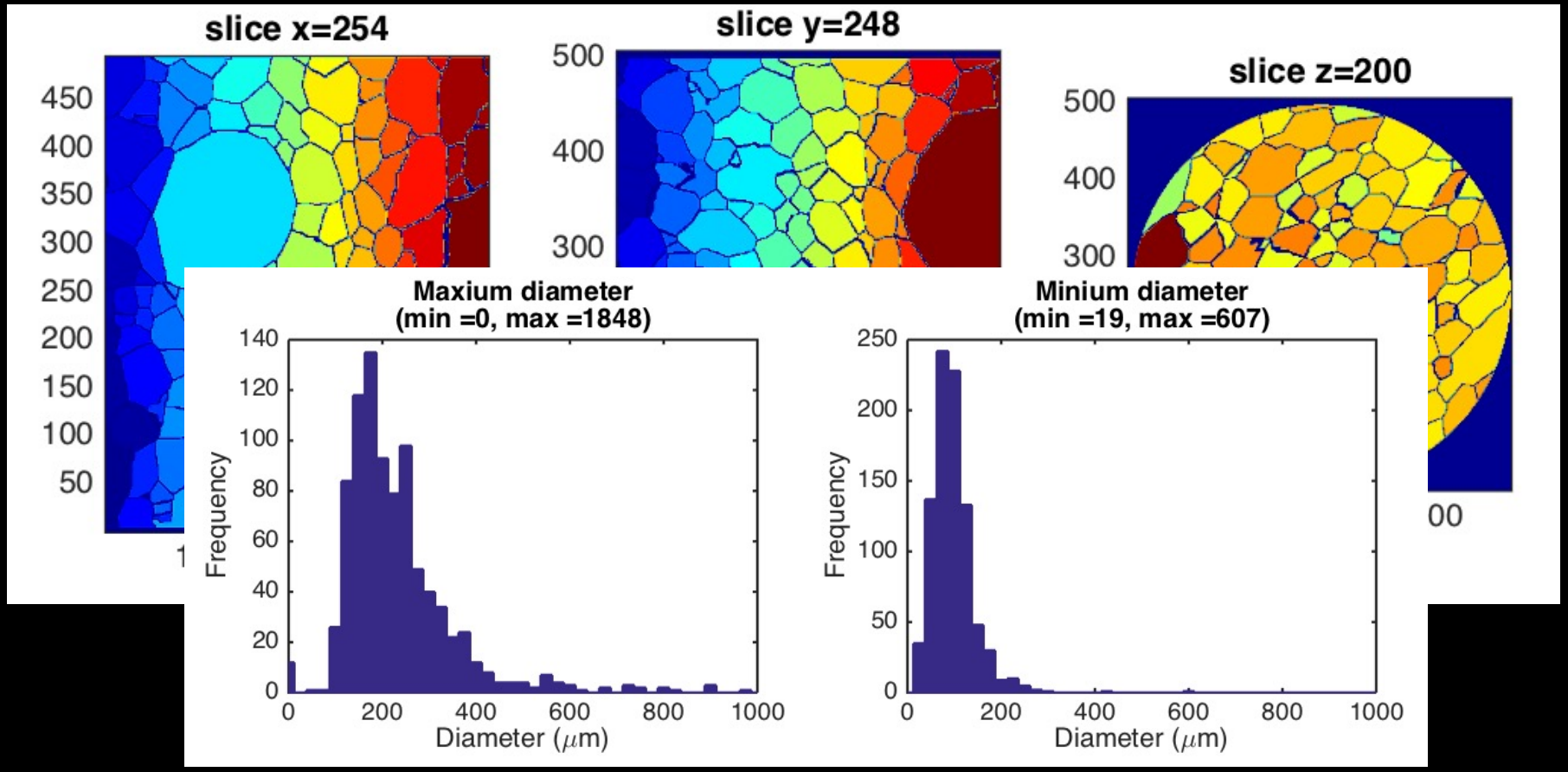
Collab. L. Bergström, S
Gordeyeva et al., 2016,

Structural imaging and characterisation: 3D image analysis, e.g.,

How do we see/identify and characterise microstructure?

Cellulose-based foam

Voxel width of 3 μm
Field of view 3 mm



Collab. L. Bergström, S
Gordeyeva et al., 2016,

First step:

Explore the image volume and visualise as for the lego example

- Using fiji and Matlab
- Create a 3D visualisation of the volume
- Investigate the grey-scale values in the volume viewer

Next: binarise, segment, label & measure foam cells

- In Matlab:
- Read in data as in lego example
- Make an image “mask” which we will use later – this is just another 3D matrix with ones in the “sample region” and zeros elsewhere:

```
mask = data;
mask(mask>0.001) = 1;
mask = uint8(mask);
```

- To understand what this is try plotting some slices

Next: binarise, segment, label & measure foam cells

- In Matlab:
- Read in data as in lego example
- Make histogram
  - How many peaks are there?
  - How many do you expect? (if this is not the same as the number of peaks you see, why do you think this is?)
  - Ignoring the peak at intensity = 0, what phase does the main peak correspond to? (you may have already worked this out in the Fiji visualisation)
- Identify the threshold that “best” separates the main phases…

  (hint: in the log(frequency) histogram of the intensities a subtle change in the slope can be seen just after the main peak, which probably indicates the best threshold value).

Binarisation and thresholding

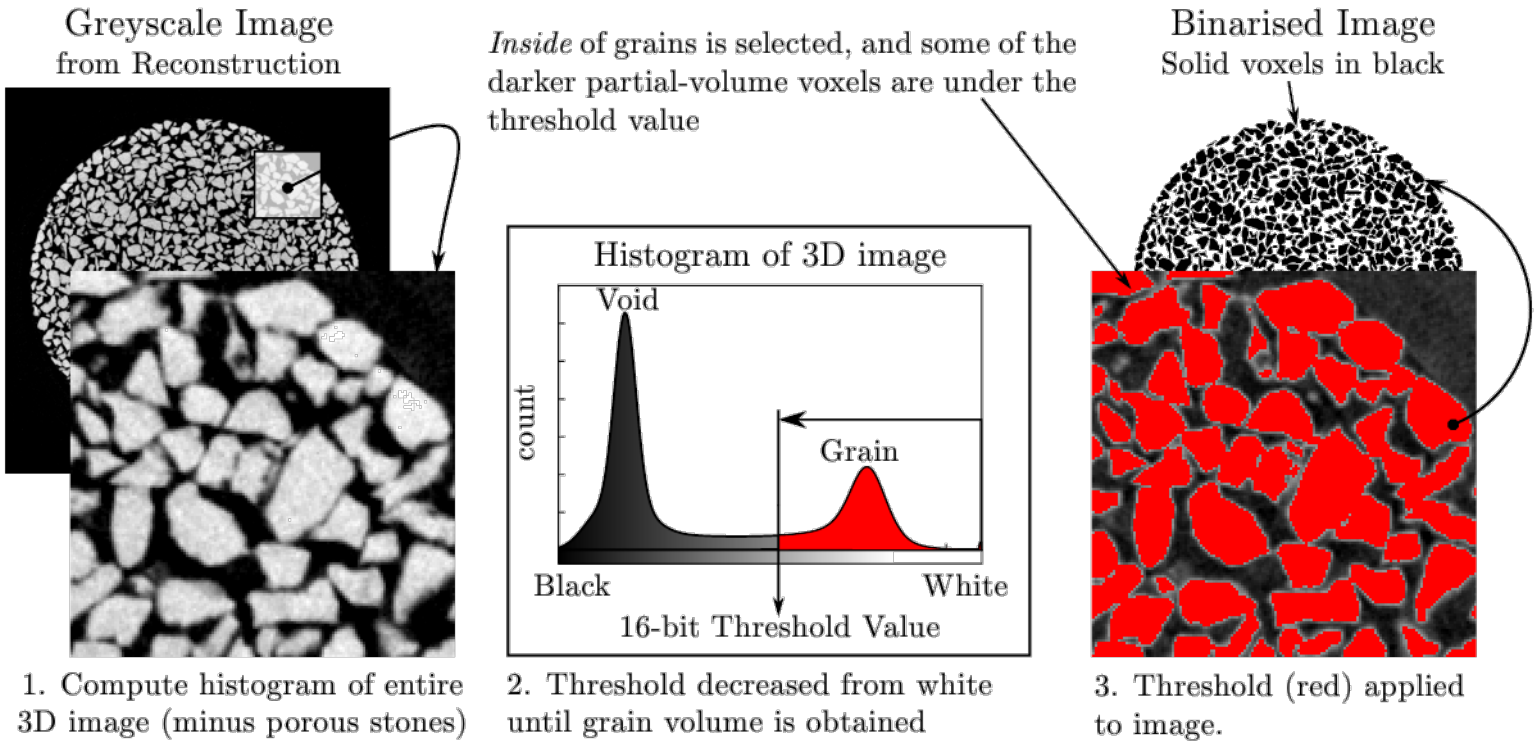


Figure 4.2: Illustration of the application and calculation of a threshold value to a slice from test HNEA03. The voxels selected by the threshold (in red) show that the darker partial volume voxels on the outsides of the grains lie below the threshold and are thus not selected.

Image from PhD E. Andò (Grenoble, 2013)

Can be extended to multiple phases, e.g., trinarisation of grain, water and air in images of partially saturated sand

Next: binarise, segment, label & measure foam cells

- Threshold as in lego example using the identified value from the histogram

```
%binarise
bin = data;
bin(bin<0.15) = 0;
bin(bin>0.15) = 1;

% change format to uint8, which makes some things easier later on
bin = uint8(bin);
```

- Make some slices through the volume image
  - Does the binarisation look OK?

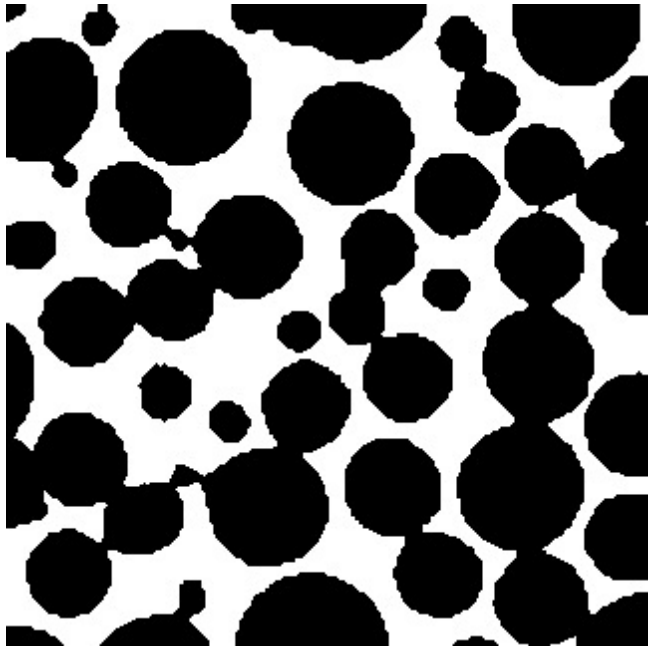
Next: binarise, segment, label & measure foam cells

- Calculate the distance map for the watershed transform (see following):

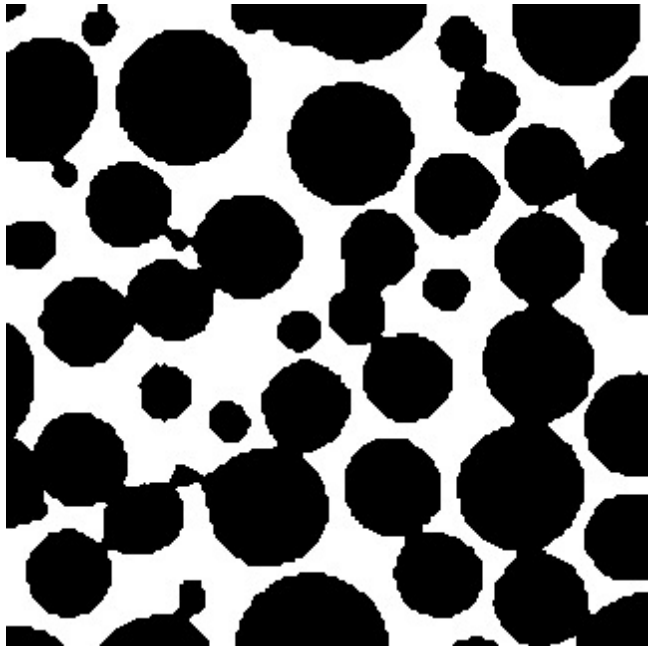
```
D1 = bwdist(bin);

% multiple the image by the “mask” to get rid of regions outside the sample

D1 = D1.*single(mask);
```

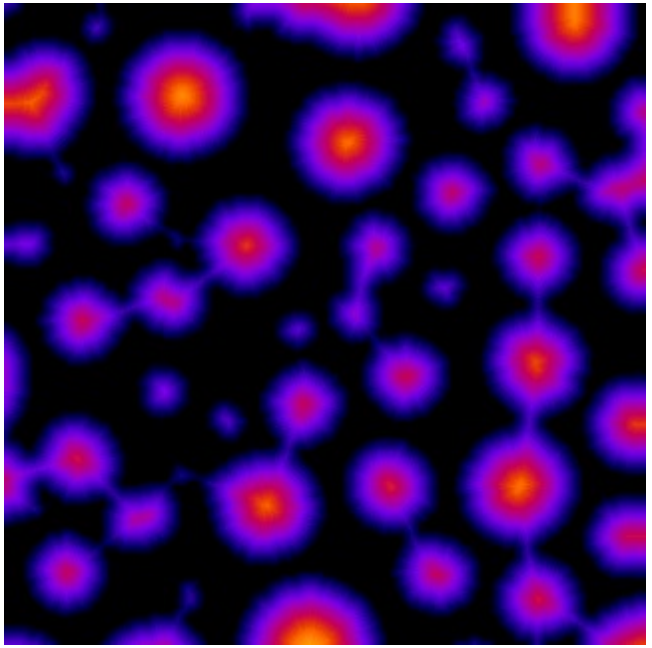
- Make some slices through the matrix – how does it look?

Watershed Segmentation - introduction of distance map



Watershed Segmentation - introduction of distance map





Watershed Segmentation - introduction of distance map

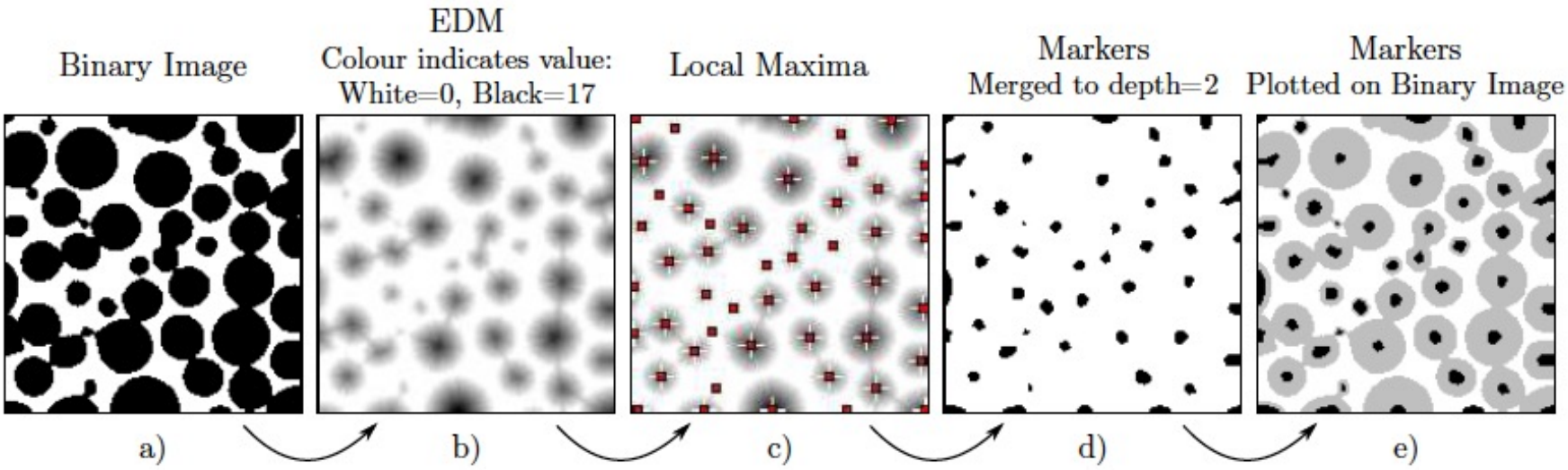


Image from PhD E. Andò (Grenoble, 2013)

Watershed Segmentation

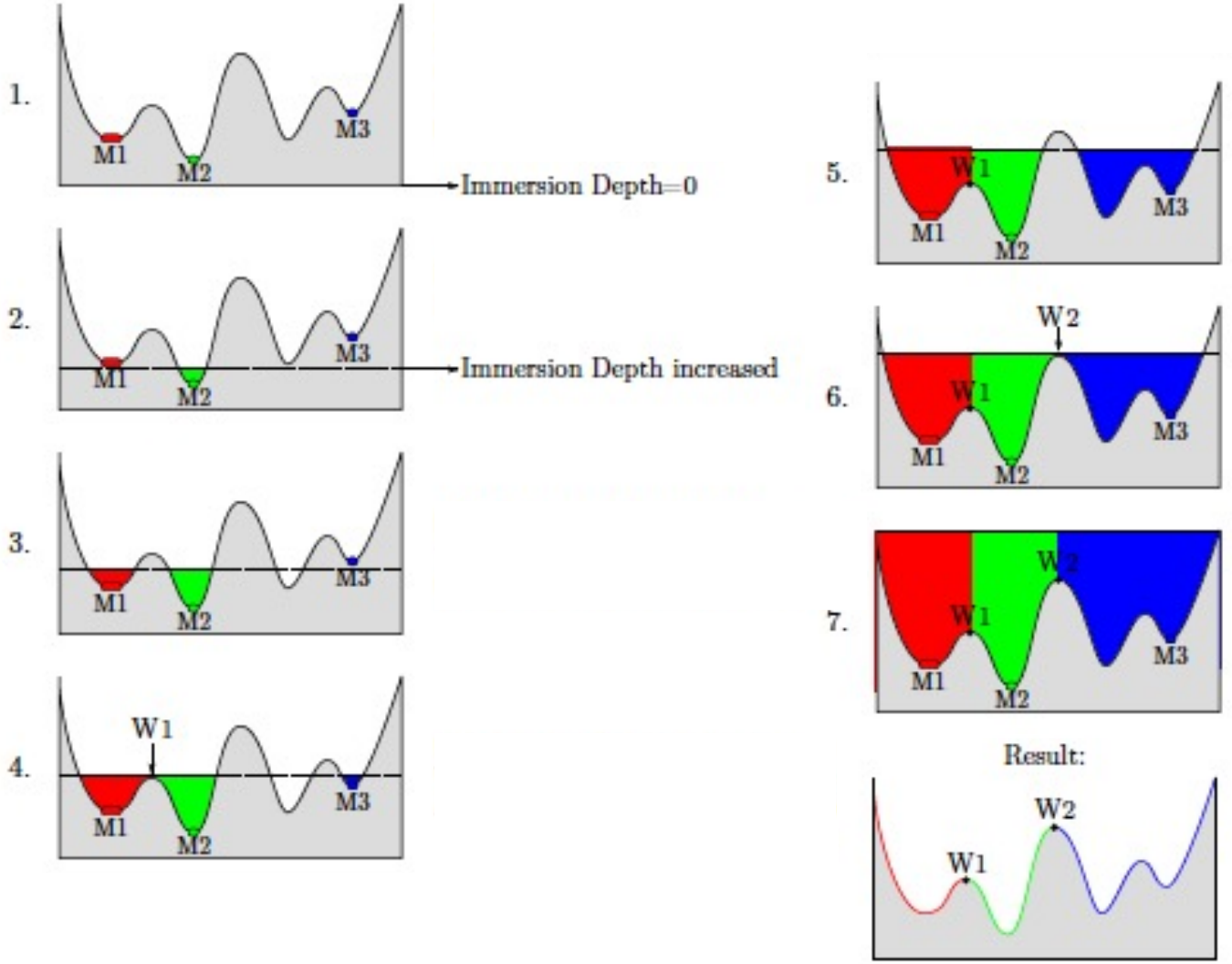


Image from PhD E. Andò (Grenoble, 2013)

Next: binarise, segment, label & measure foam cells

- Perform the watershed transform on the distance map:

```
L = watershed(-D1);

% use the mask to only take labels in the sample area

labels = L.*uint16(mask);
```

How does this look? Does it make sense in terms of identifying and separating the foam cells?

Next: binarise, segment, label & measure foam cells

- How does this look? Does it make sense in terms of identifying and separating the foam cells?

… probably not

… the binary image probably had a number of “extra” points that were not in the cell walls, which we need to remove so that the distance map and watershed are “correct”

So we will redo the last few steps after the binarisation, with some additions…

Next: binarise, segment, label & measure foam cells

… the binary image probably had a number of “extra” points that were not in the cell walls, which we need to remove so that the distance map and watershed are “correct”

```
%binarise
bin = data;
bin(bin<threshold) = 0;
bin(bin>threshold) = 1;

bin = uint8(bin);

% get rid of isolated points in binary
bin2 = bwareaopen(bin,4500);

%distance map

D1 = bwdist(bin2);

D1 = D1.*single(mask);

% filter out local maxima

D2 = imhmax(D1,5);

L = watershed(-D2);

labels = L.*uint16(mask);
```

Check the help on each command to find out what these are doing so you can explain it!

Next: binarise, segment, label & measure foam cells

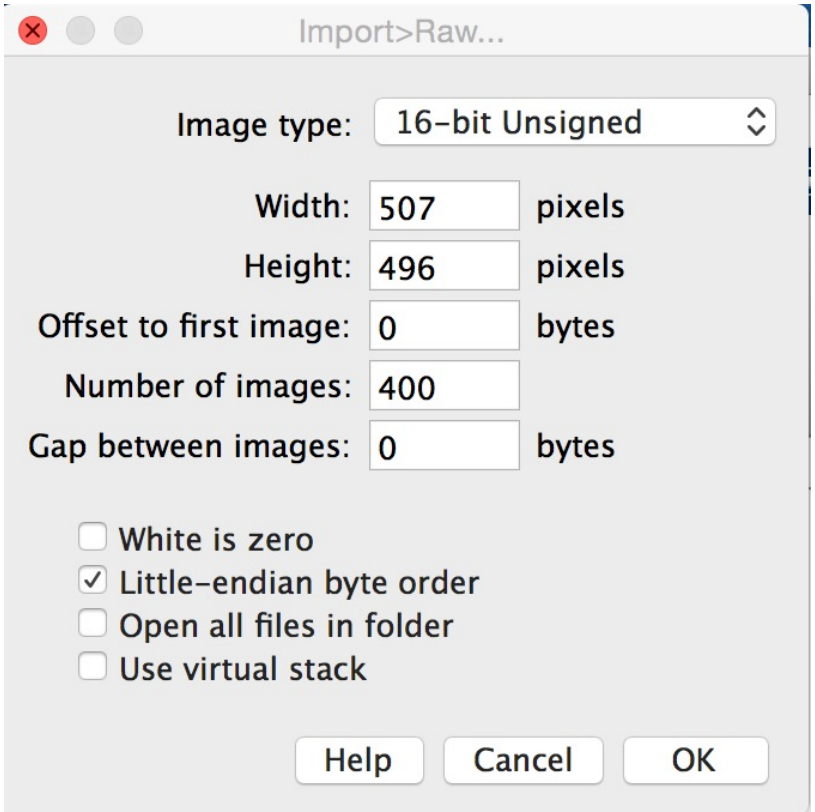
You can check the result by making slices through the volume, or using the “volumeViewer” in matlab
(https://se.mathworks.com/help/images/ref/volumeviewer-app.html)

or could output it to view in Fiji….

To write out a binary file:

```
fid = fopen('labels_16bit.raw','w');
fwrite(fid,labels,'uint16');
fclose(fid);
```

Read it into Fiji with:
file > import > raw



Next: binarise, segment, label & measure foam cells

Now do some measurements using “regionprops3” (new in R2017b)

```
% count number of pixels in each label (volume):

stats = regionprops3(labels, 'SurfaceArea',’Volume’);

volume = cat(1, stats.Volume);

surfaceArea = cat(1, stats.SurfaceArea);
```

- You can measure other structural parameters – check out the “regionprops3” manual page
- Now make histograms of the results
- Perhaps also calculate “sphericity” (look this up on wikipedia)
- Make sure you can follow (and describe) what has been done here.