ImageJ: a useful image reprocessing tool

Contents

- Remeasure distance on an SEM (JPG)
- Calculate open area
- Construct 3D images

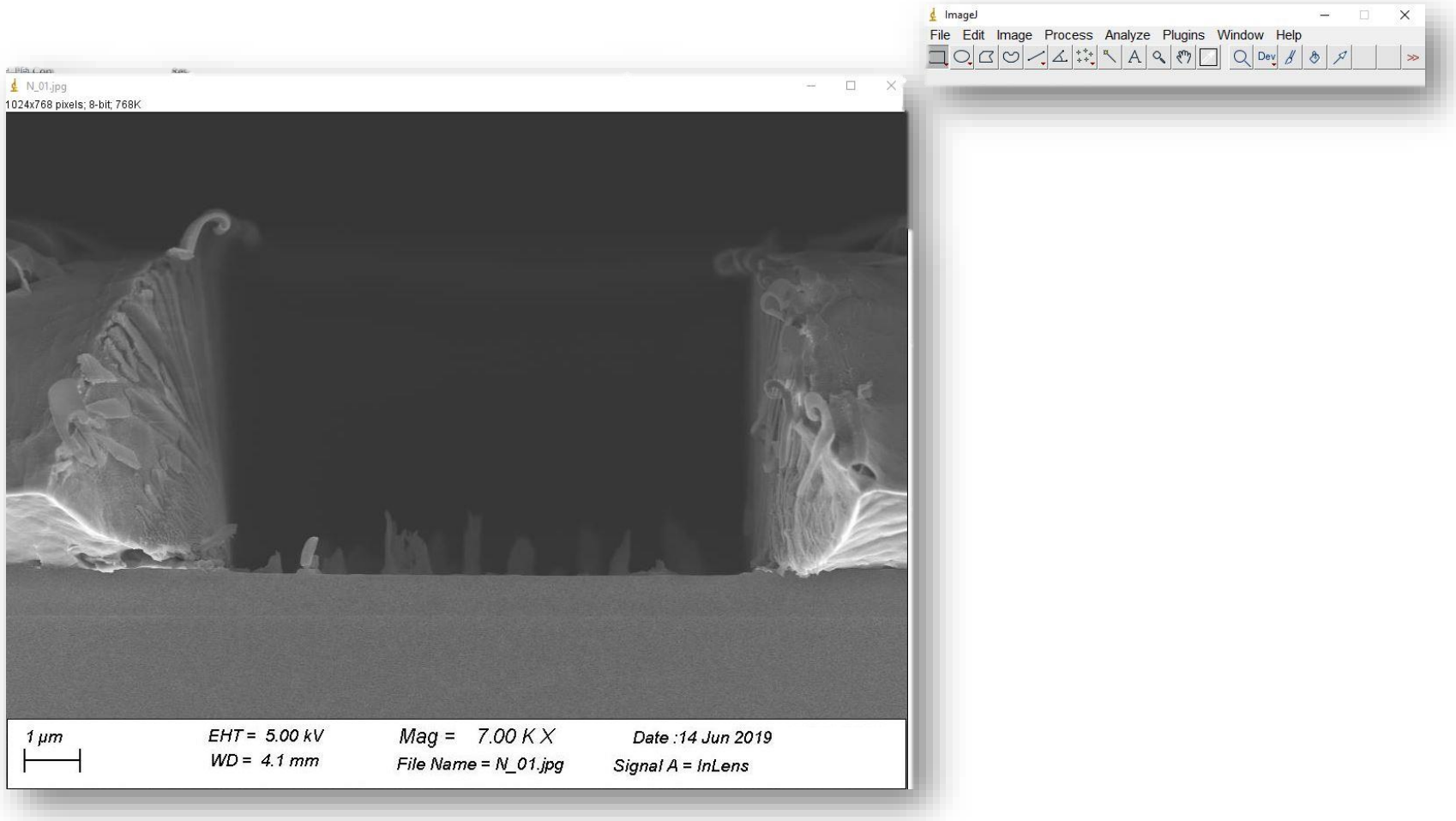
Remeasure an SEM image

- Download from https://imagej.nih.gov/ij/download.html

- Alternative versions available like Fuji.

Remeasure an SEM image

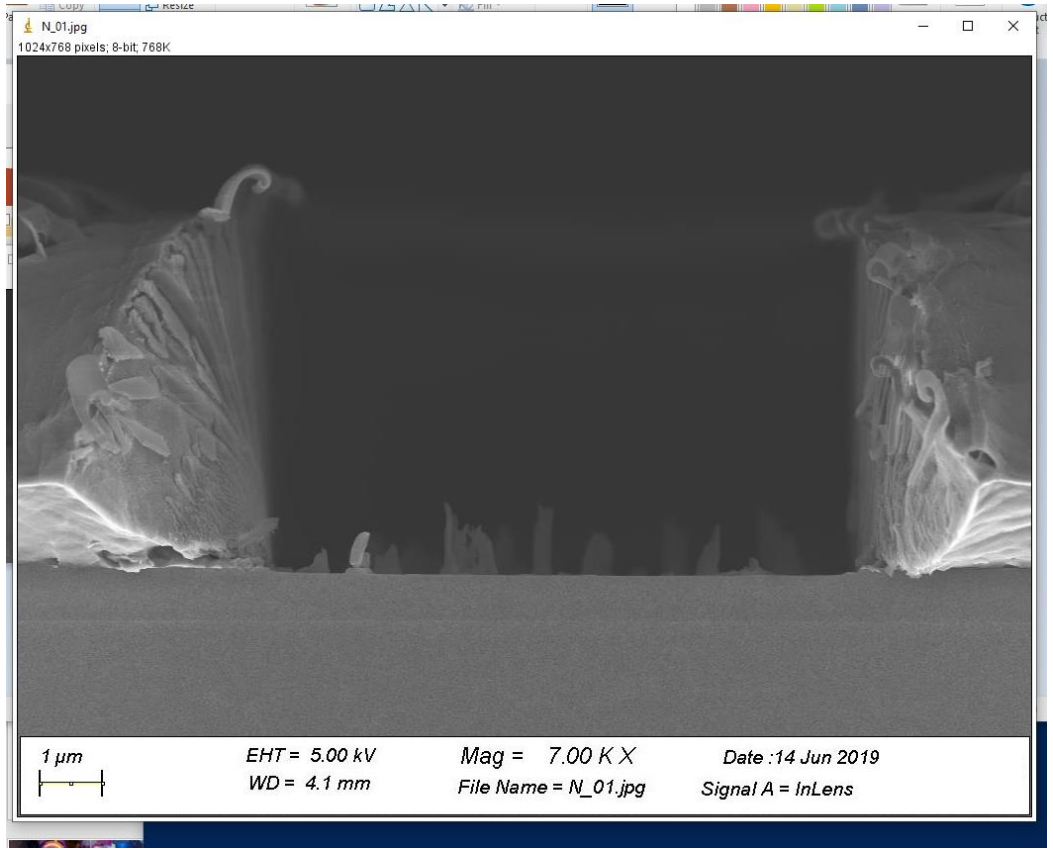
Open an image



Remeasure an SEM image

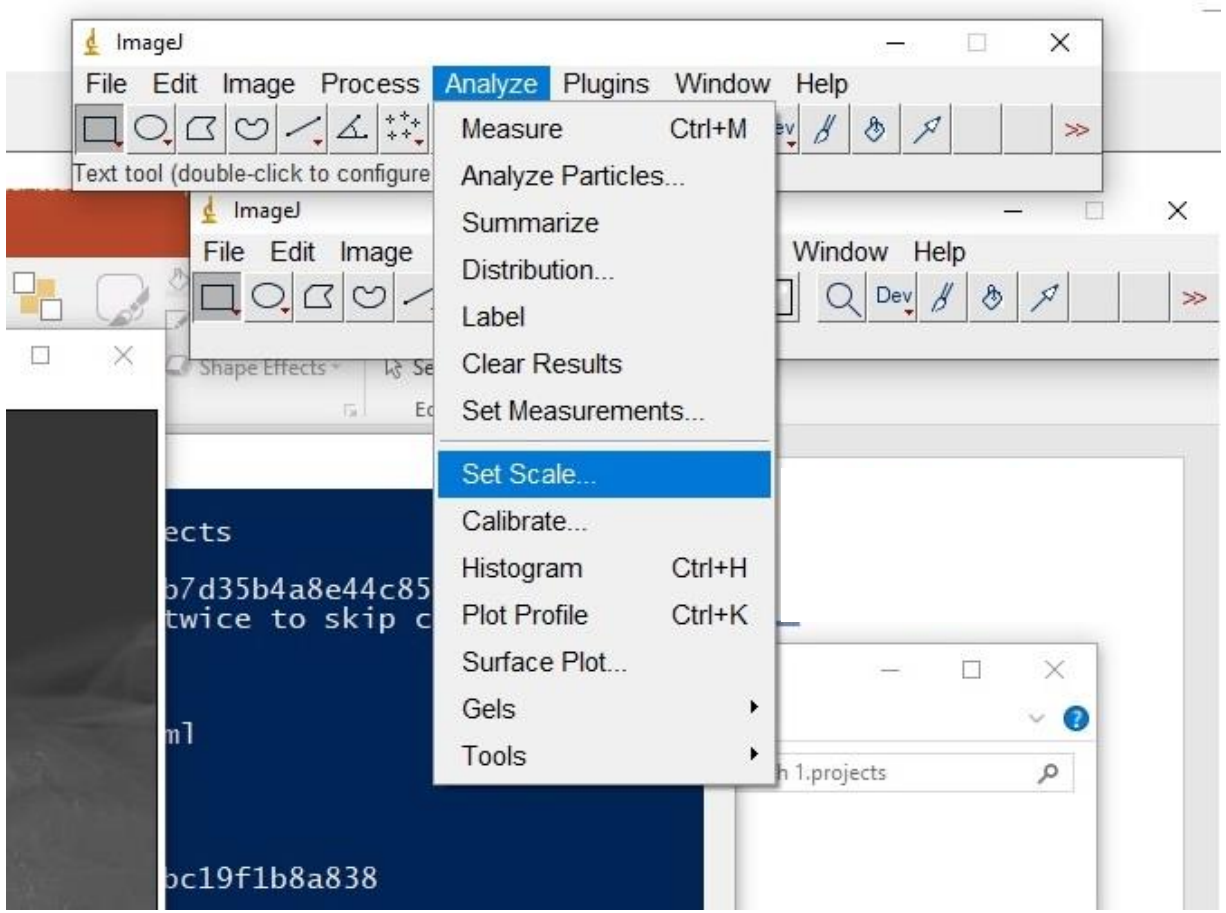
Use `line' tool to draw over the scale

Hold shift for a horizontal line



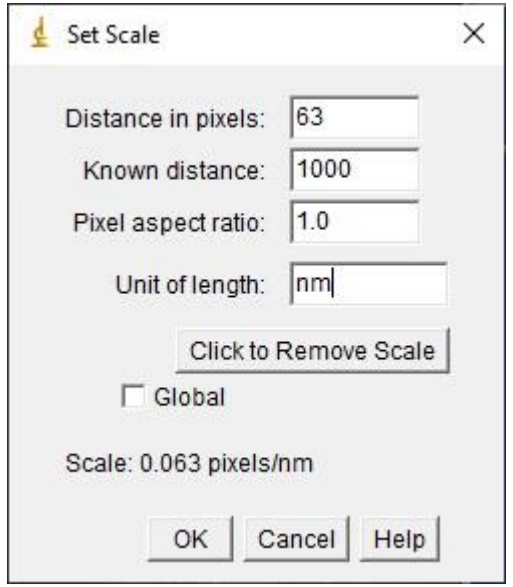
Remeasure an SEM image

Analyze -> Set Scale…



Remeasure an SEM image

Tell ImageJ what the known scale length is



Remeasure an SEM image

OR, even better! Just use the AutoScale marco!

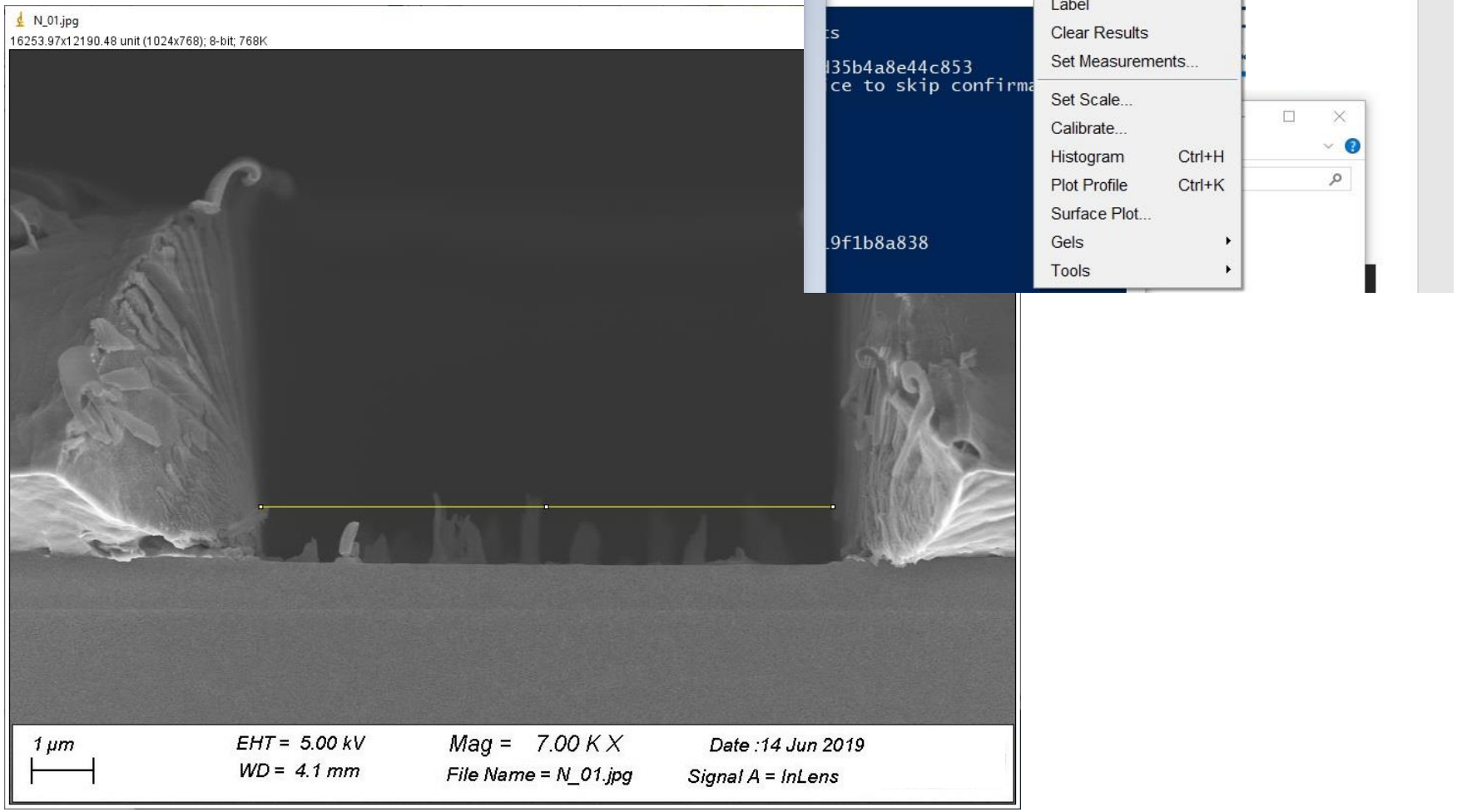
It finds the scale mark, reads the text, set it up for you!

The only known issue: can sometimes tell you the unit is `Âµm’ rather than `µm’…
This is because `µm’ in Unicode is `Âµm’ in UTF-8
So check your encoding…

```
////////////////////// begin macro
requires( "1.51t" );
run( "8-bit" );
b = 10;
setScaleFromBar( b );
exit();
////////////////////// end macro
////////////////////// begin functions
function diff( aa ) {
                        for ( i = 1; i < aa.length; i++ ) {
                                                aa[i-1] = aa[i] - aa[i-1];
                        }
                        aa[aa.length-1] = 0;
}
function getBeginEnd( aaa ) {
                        mi = Array.findMinima( aaa, 0 );
                        mx = Array.findMaxima( aaa, 0 );
                        if ( mx[0] < mi[0] ) aaa[1] = mx[1]; else aaa[1] = mx[0];
                        if ( mi[0] > aaa[1] ) aaa[0] = mi[1]; else aaa[0] = mi[0];
}
function setScaleFromBar( bb ) {
                        run( "Set Scale...", "distance=0 known=0 pixel=1 unit=pixel" );
                        ww = getWidth();
                        makeRectangle( 0, bb, ww, getHeight() - 2 * bb );
                        setKeyDown( "alt" );
                        a = getProfile();
                        setKeyDown( "none" );
                        min = Array.findMinima( a, 0 );
                        if ( min[0] > min[1] ) yy = min[1]; else yy = min[0];
                        pel = abs( min[1] - min[0] );
                        a = Array.slice( a, yy + pel );
                        diff( a );
                        getBeginEnd( a );
                        makeRectangle( 0, yy + bb + pel + a[0], ww, abs( a[0] - a[1] ) );
                        a = getProfile();
                        run( "Select None" );
                        diff( a );
                        getBeginEnd( a );
                        pel = abs( a[0] - a[1] );
                        run( "Set Scale...", "distance=" + pel + " known=500 unit=µm" );
}
////////////////////// end functions
```

Remeasure an SEM image

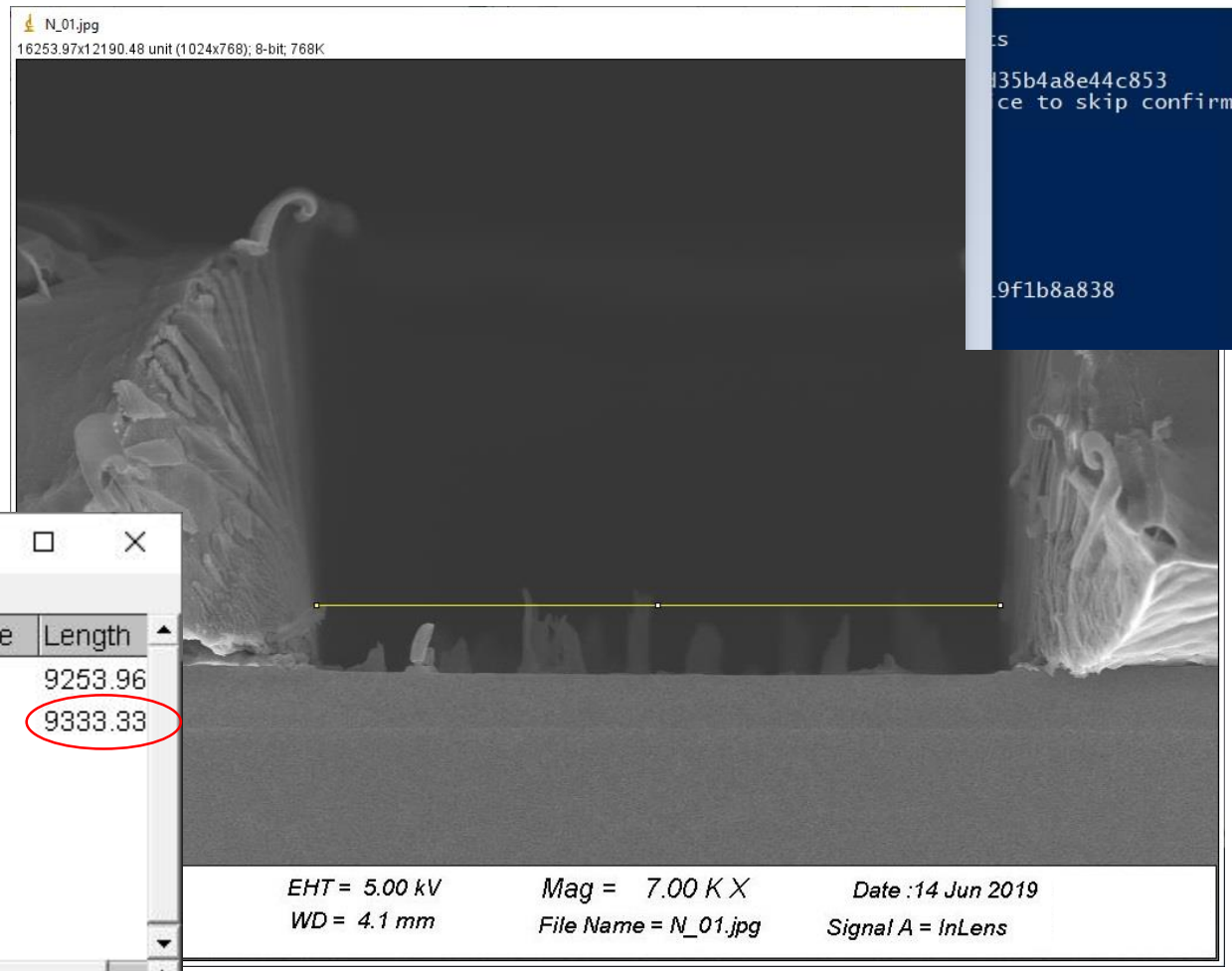
Measure a gap of your choice

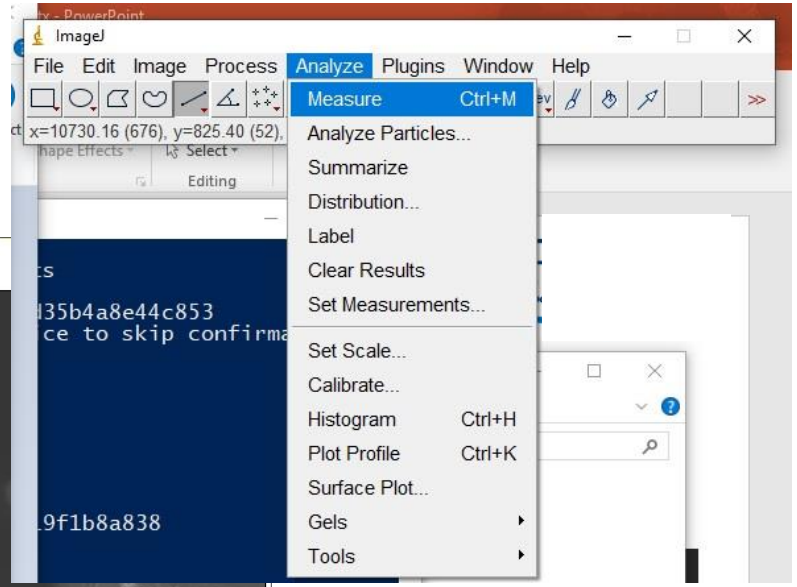


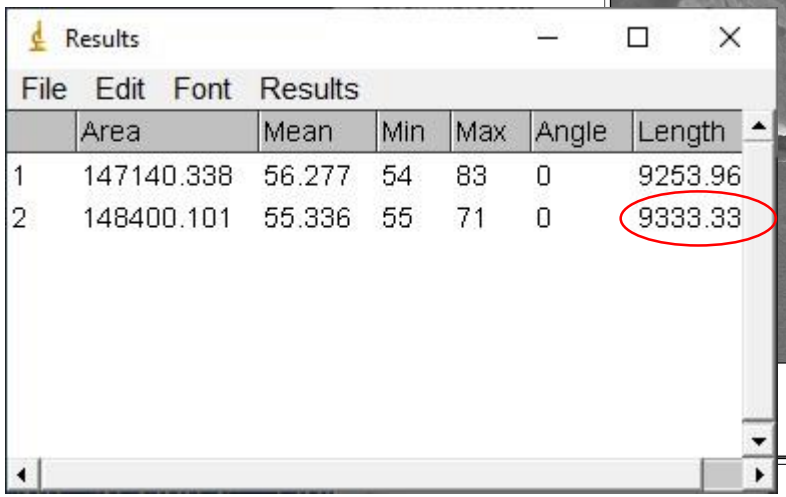
Remeasure an SEM image

Measure a gap of your choice

Then Ctrl+M







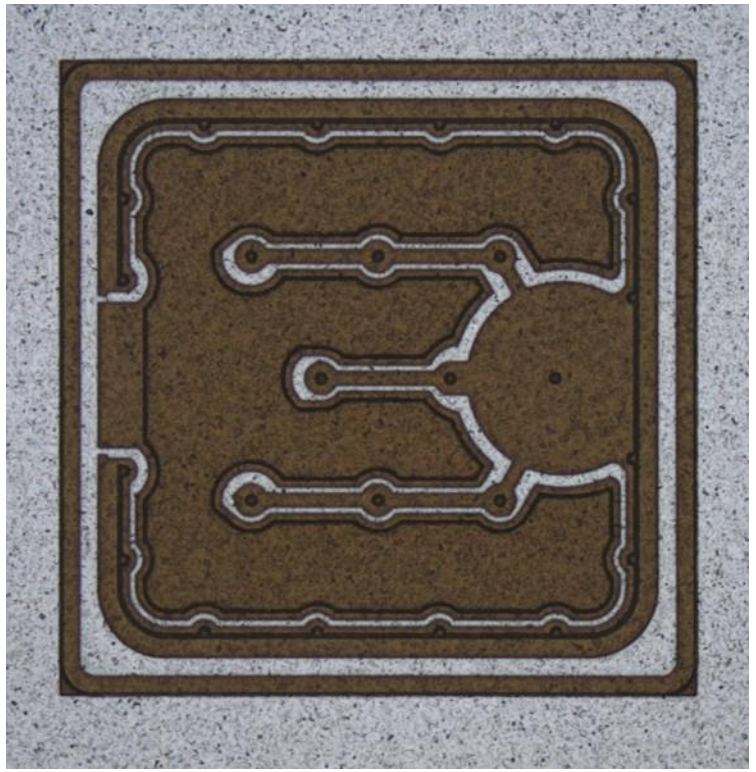
	Area	Mean	Min	Max	Angle	Length
1	147140.338	56.277	54	83	0	9253.96
2	148400.101	55.336	55	71	0	9333.33

Calculate open area

Calculate open area

Customer told us the OA is 40%

We are not convinced…

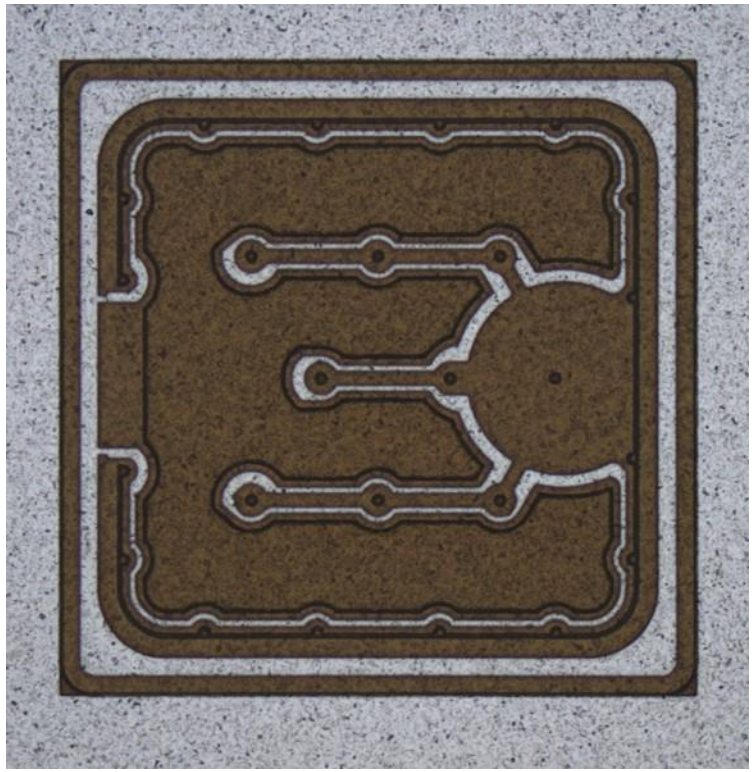


Calculate open area

Name the file `1.jpg’

Put it in the same folder as `OA.exe’ (or `OA.py’ if you have all the libs installed)

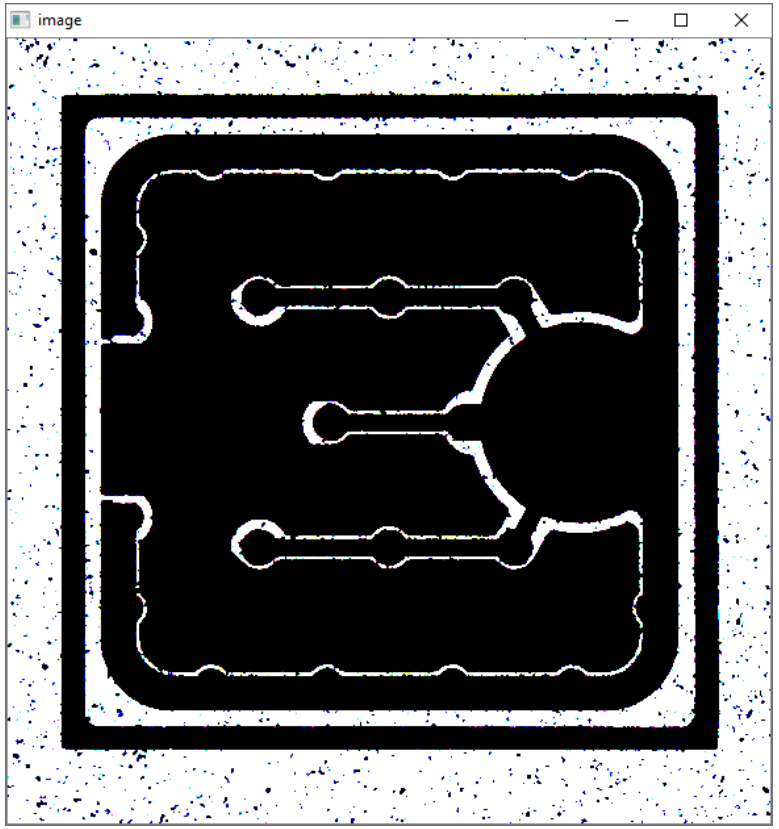
OA.exe is available my SEM folder



Calculate open area

Run OA.exe

A window will pop up showing the processed binary image

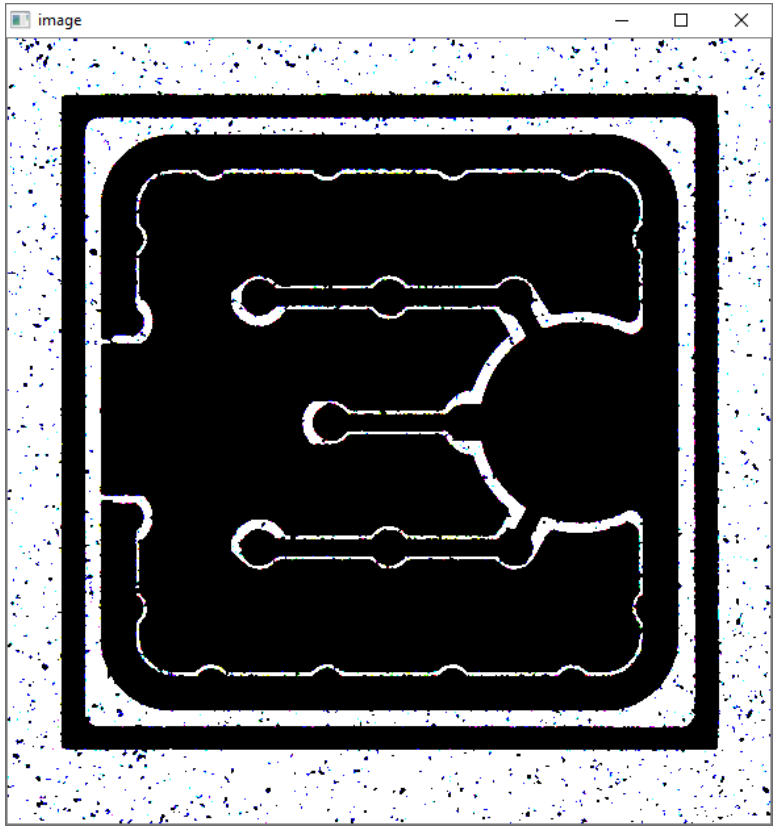


Calculate open area

Run OA.exe in cmd/powershell

A window will pop up showing the processed binary image

Command line will tell you `Open area is 37.69%’

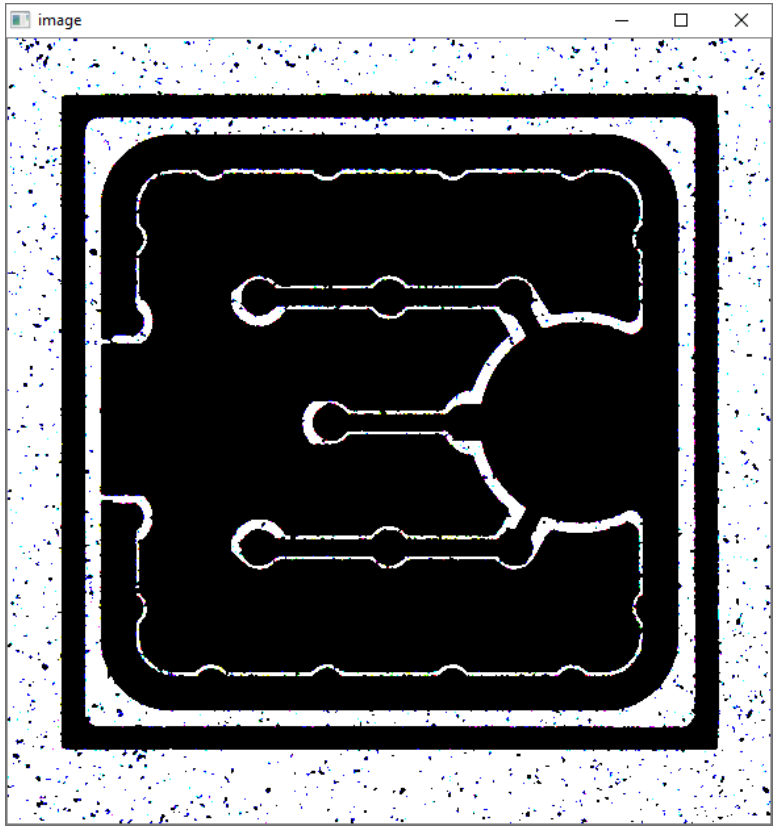


Calculate open area

Alternatively, use ImageJ

Binarise the image, denoise it by using remove outliers
Find the result in Image-Adjust-Threshold

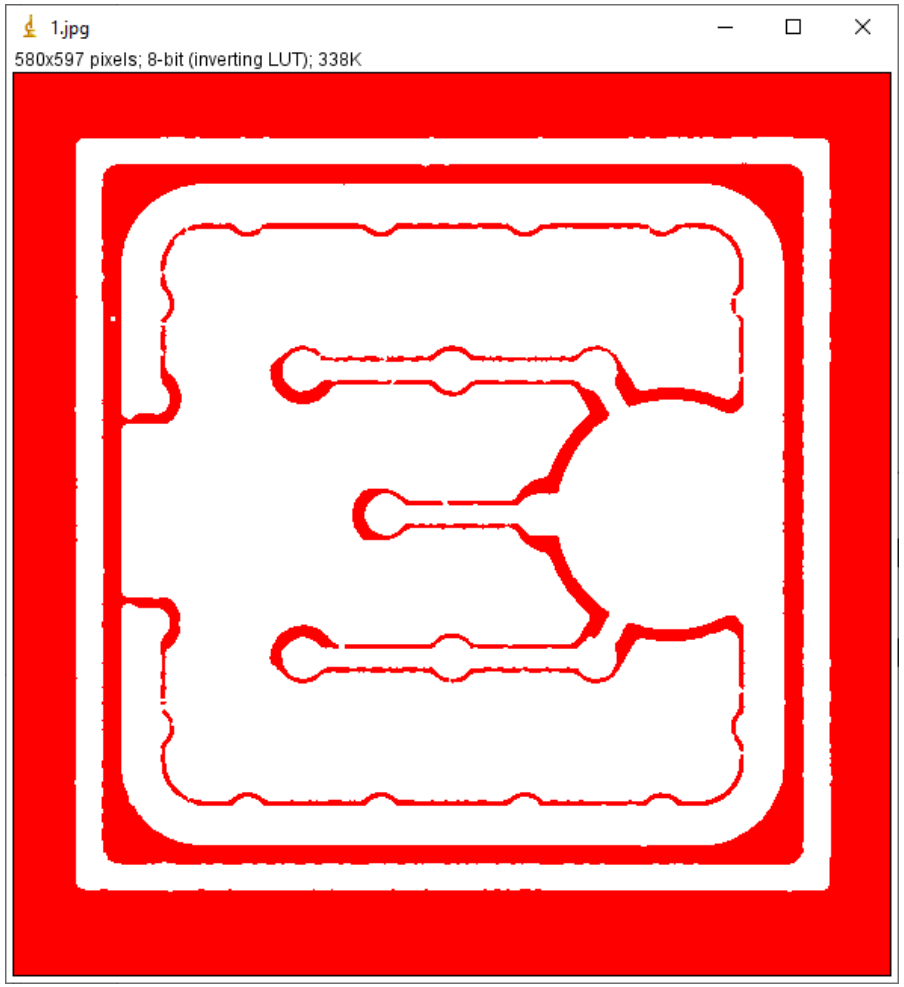
Result is 39.03% (thx to denoise algorithm)



Calculate open area

Marco here:

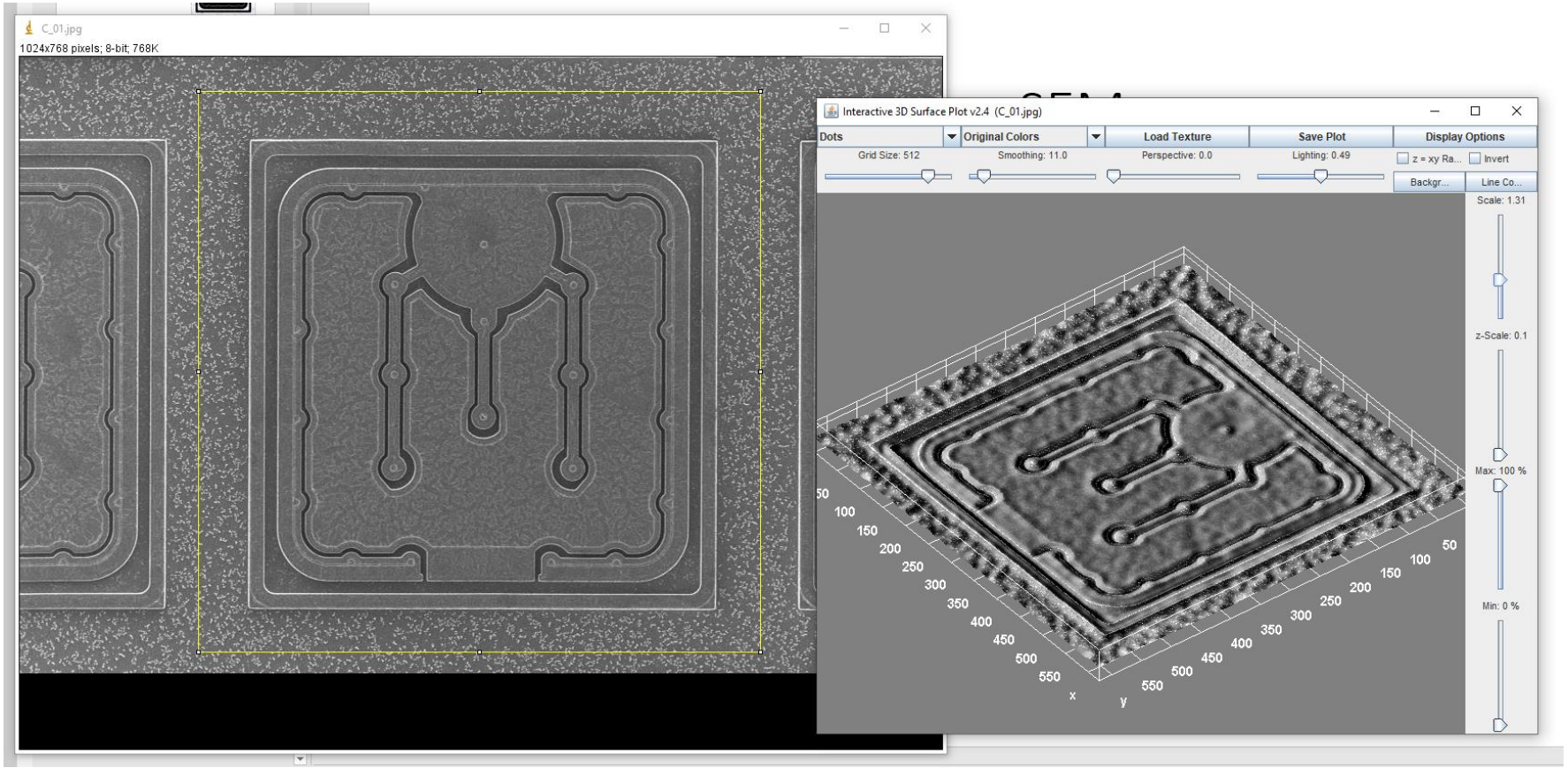
```
setOption("BlackBackground", false);
run("Make Binary");
run("Remove Outliers...", "radius=2 threshold=50
which=Bright");
run("Remove Outliers...", "radius=2 threshold=50
which=Bright");
setAutoThreshold("Default");
//run("Threshold...");
```



Construct 3D images

Construct 3D images – single SEM

This method reconstruct 3D image using the greyscale info from a single SEM image.
Simple use plugins->3D-> Interactive surface plot



Construct 3D images - Stack

This is yet another method to give you beautiful 3D images

First take a `stack’ of images. Here im using an optical scope as an example.

I take a photo with increasing Z. This gives photos of different focusing. Ideally the Z should be equally spaced.

Use File->Import ->Image Sequence

Construct 3D images - Stack

Use plugins->3D->Stack 3D viewer
Example taken from my thesis…