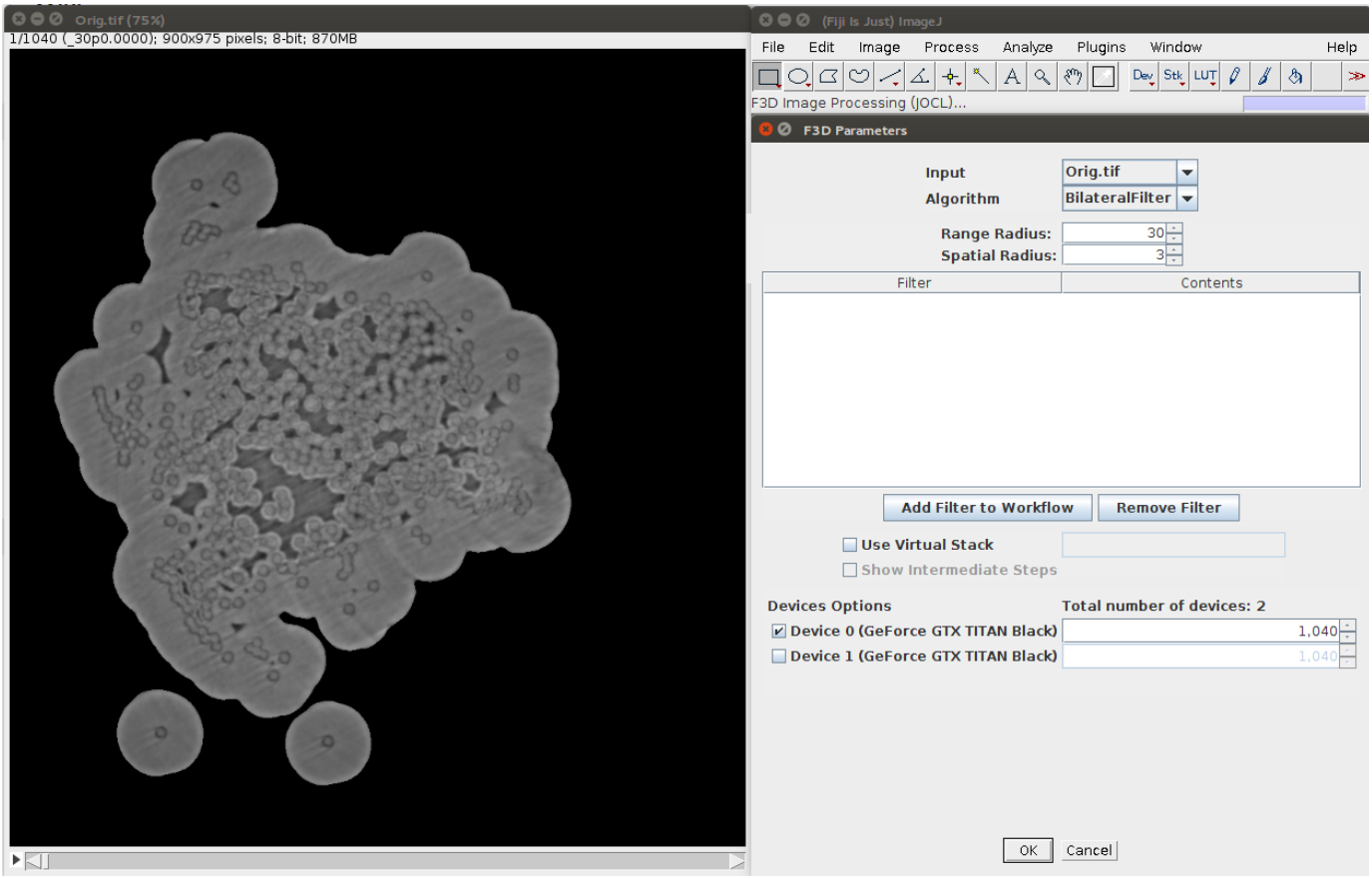
F3D Image Processing and Analysis

What is F3D?

F3D is a Fiji plugin, designed for high-resolution 3D image, and written in OpenCL. F3D plugin achieves platform-portable parallelism on modern multi-core CPUs and many-core GPUs. The interface and mechanisms to access F3D accelerated kernes are written in Java to be fully integrated with other tools available within Fiji/ImageJ. F3D delivers several key image-processing algorithms necessary to remove artifacts from micro-tomography data. The algorithms consist of data parallel aware filters that can efficiently utilize resources and can process data out of core and scale efficiently across multiple accelerators. Optimized for data parallel filters, F3D streams data out of core to efficiently manage resources, such as memory, over complex execution sequence of filters. This has greatly expedited several scientific workflows dealing with high-resolution images. F3D preforms two main types of 3D image processing operations: non-linear filtering, such as bilateral and median filtering, and morphological operators (MM) with varying 3D structuring elements.



See documentation here!

Documentation

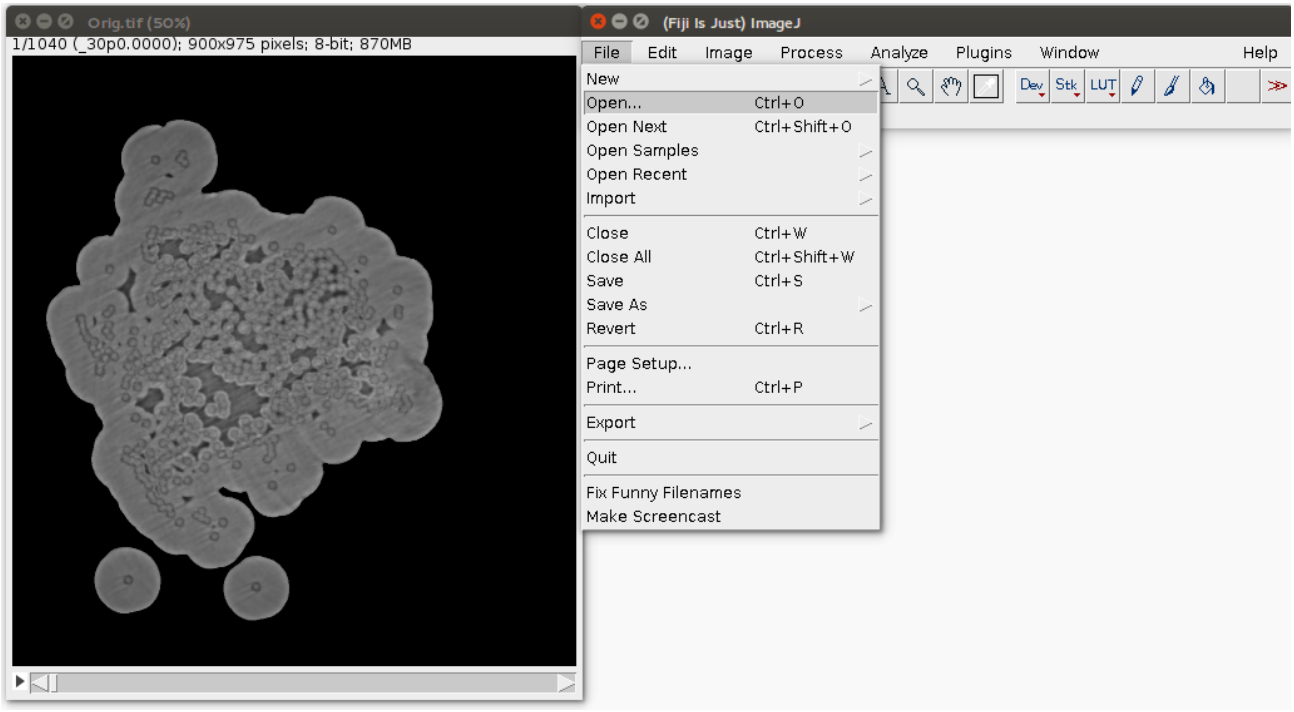
Examples index

Applying a single filter to an image using F3D

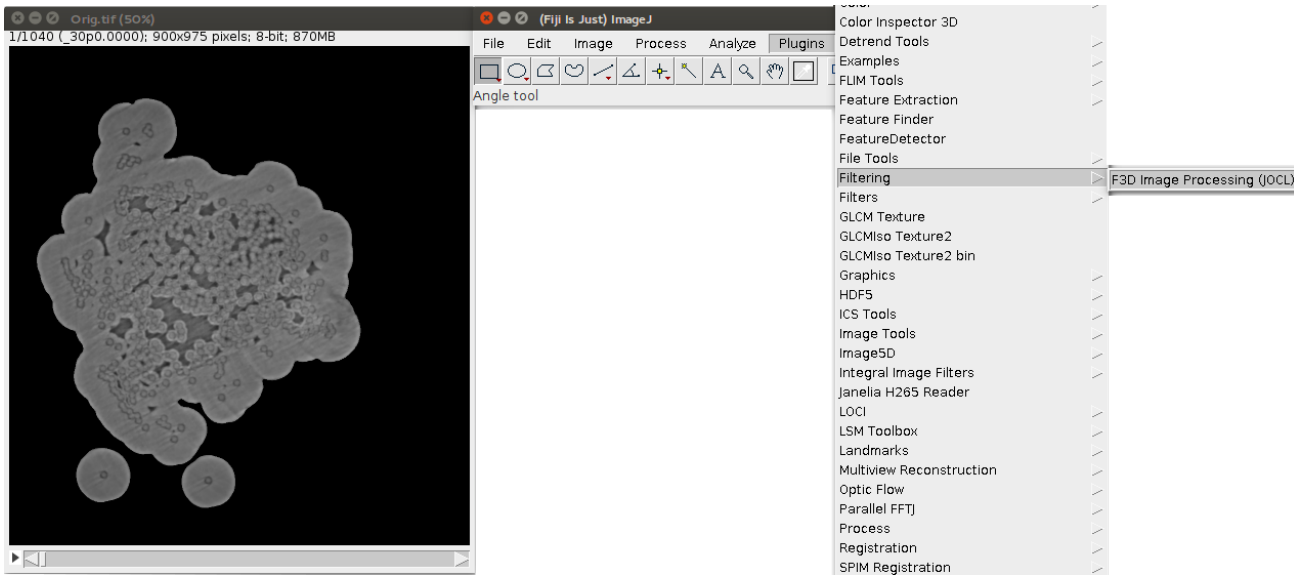
Opening the image and running the filter using the plugin GUI

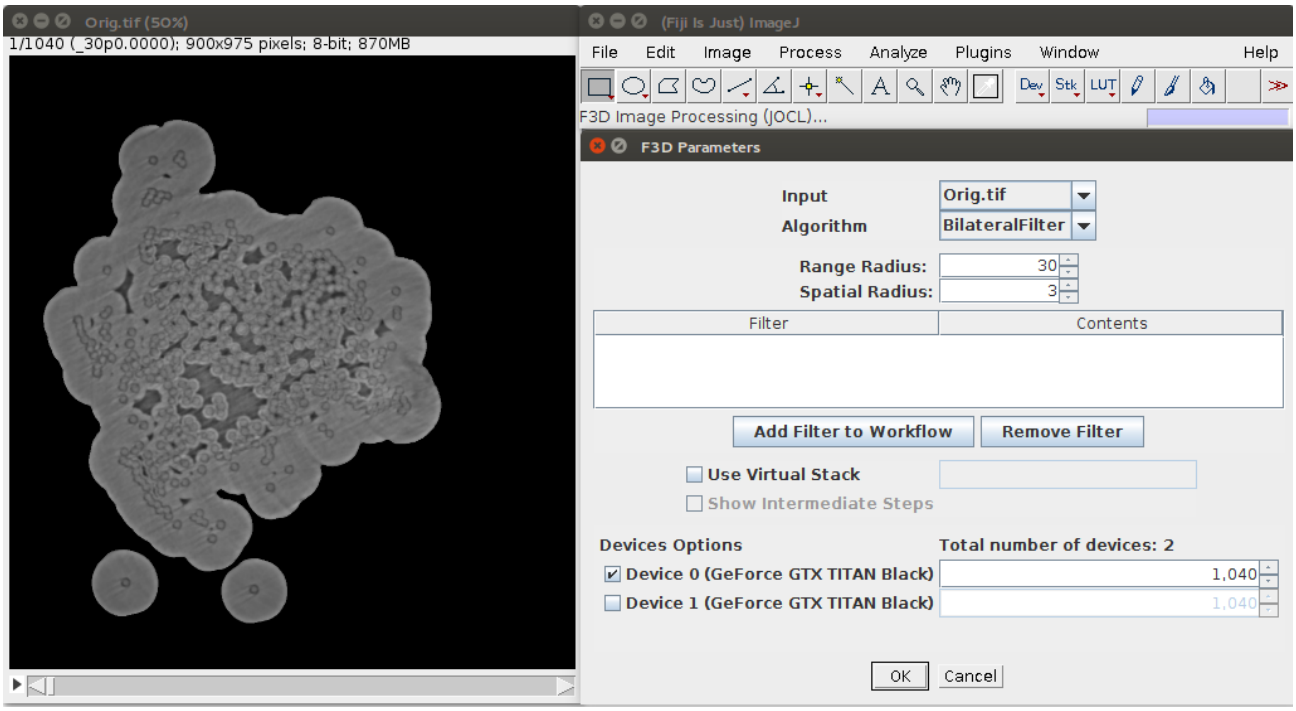
Using F3D to apply a single filter to a specific image can be done following the steps below:

1. **Open the image of interest**

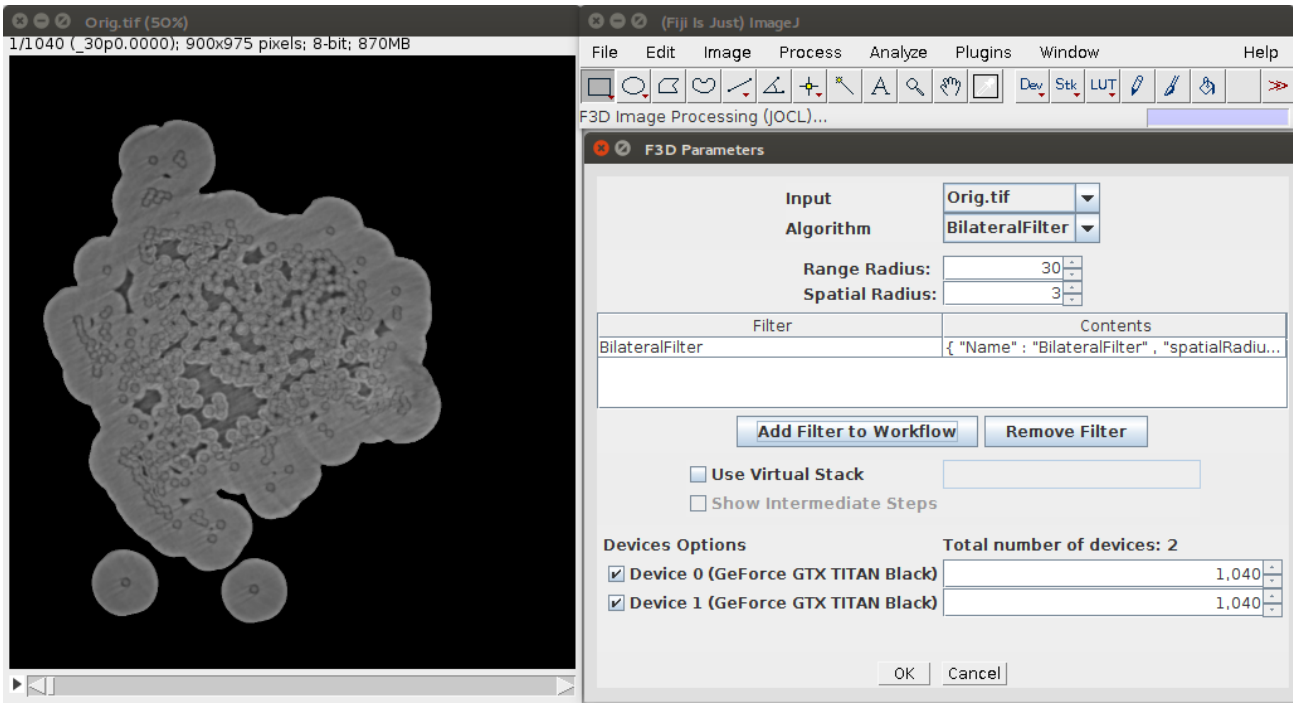


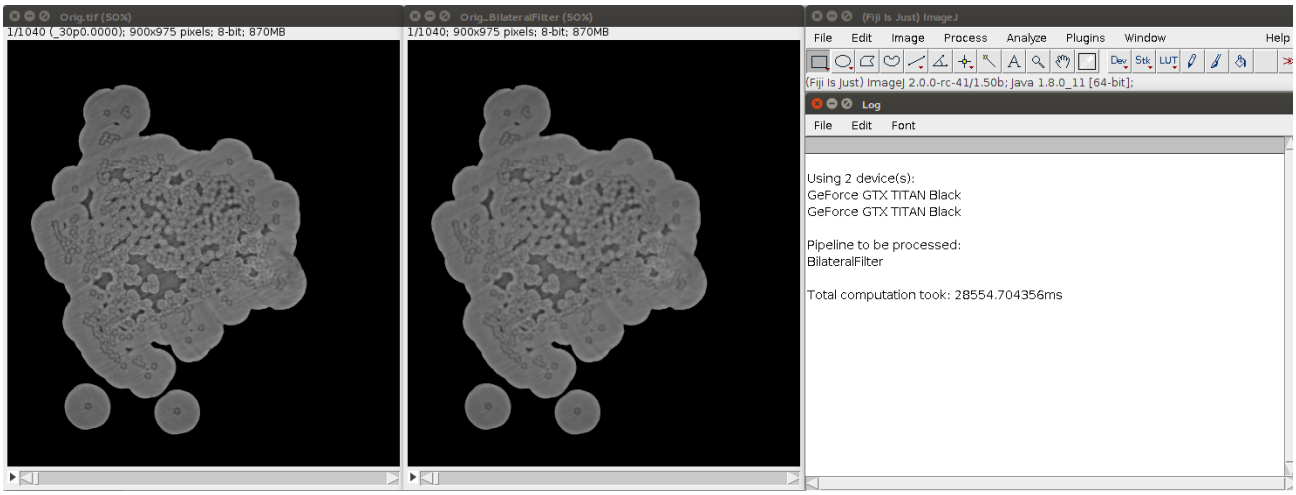
2. **Open F3D plugin**





3. **Set parameters and run the filter**





Opening the image and running the filter as an ImageJ macro

Run filter

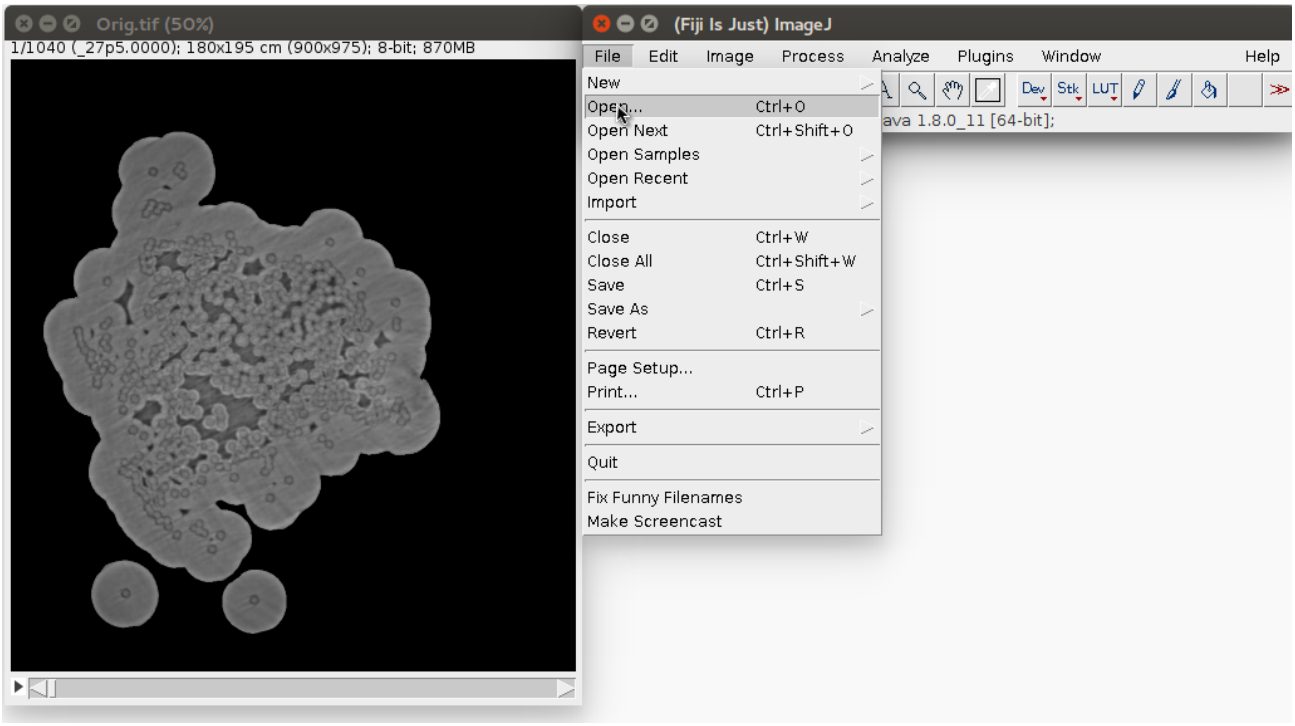
```
//Macro to open an image and apply Bilateral filter using F3D plugin
open("Orig.tif");
run("F3D Image Processing (JOCL)", "options=[{input : Orig.tif ,
intermediateSteps : false , devices : [{ Device: 0 , MaxNumSlices: 1040 } {
Device: 1 , MaxNumSlices: 1040 } ] , useVirtualStack : false ,
virtualDirectory : empty , filters : [{ Name : BilateralFilter ,
spatialRadius : 3 , rangeRadius : 100 }]}]");
```

Improving contrast of an image

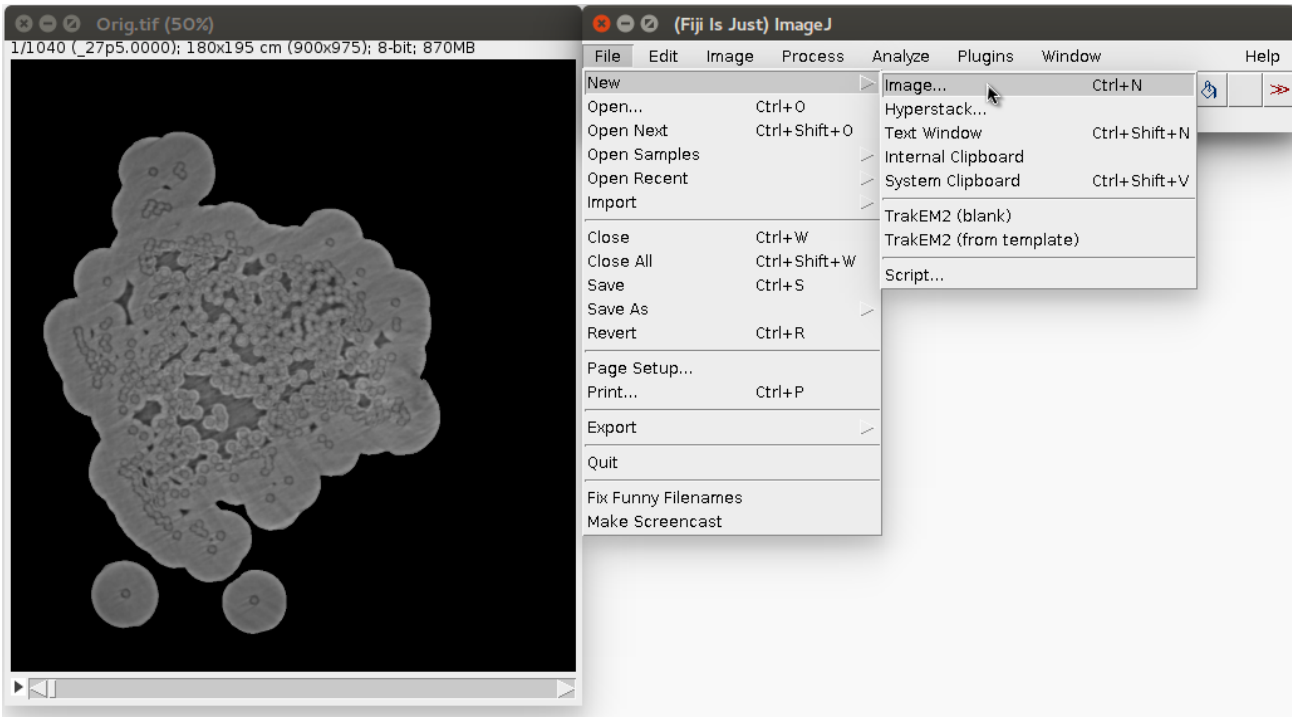
Using the GUI

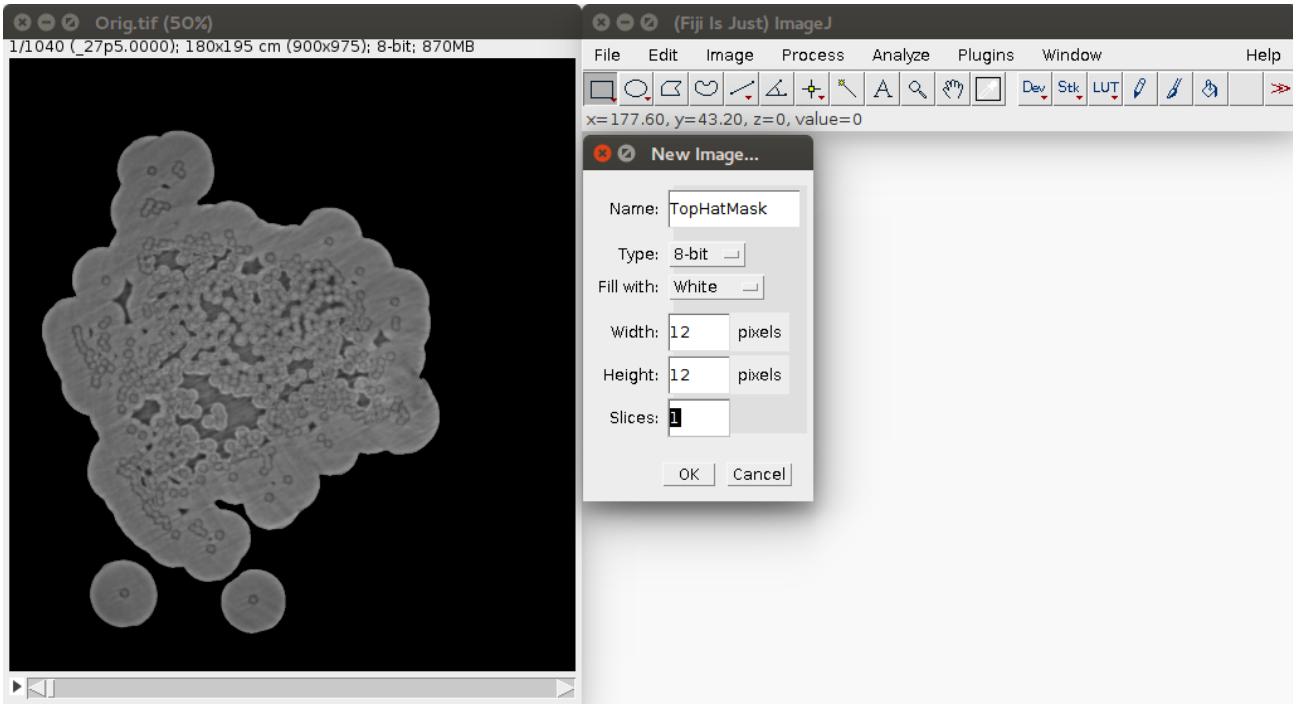
This examples show how to use a custom structure element to improve the contrast of an image:

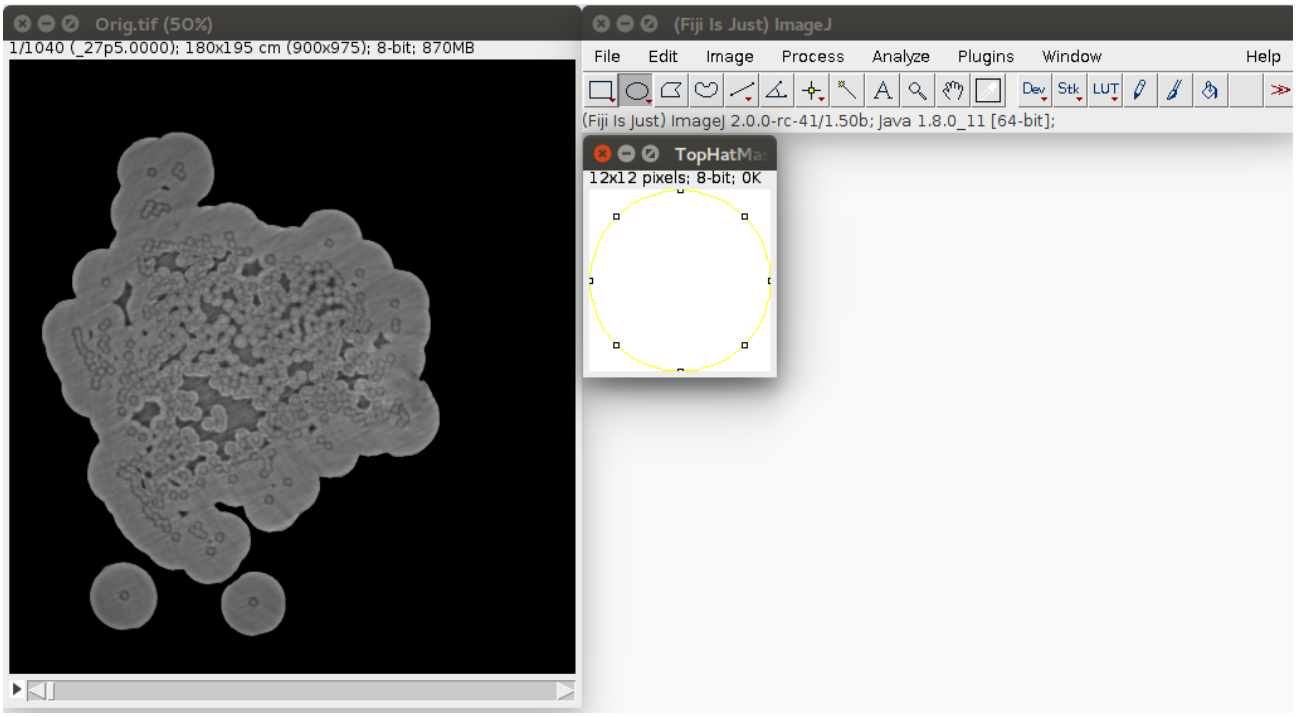
1. **Open the image of interest**

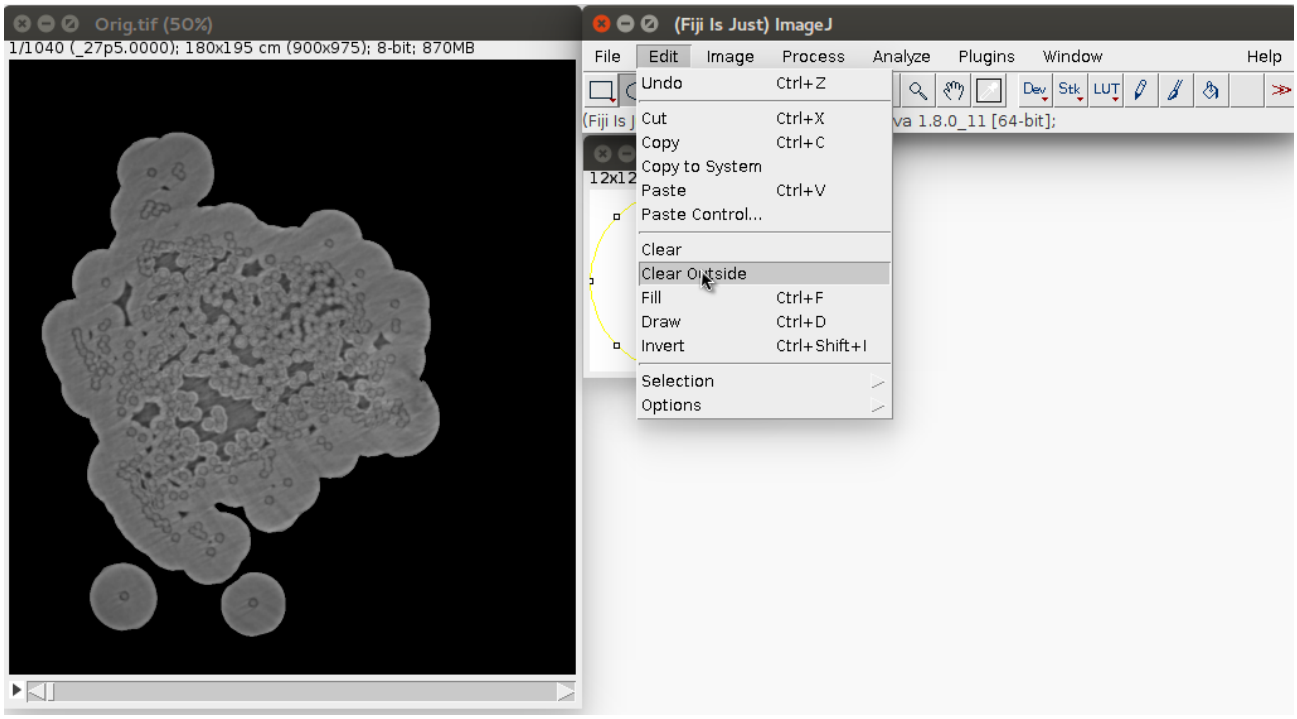


2. **Create a custom mask:**

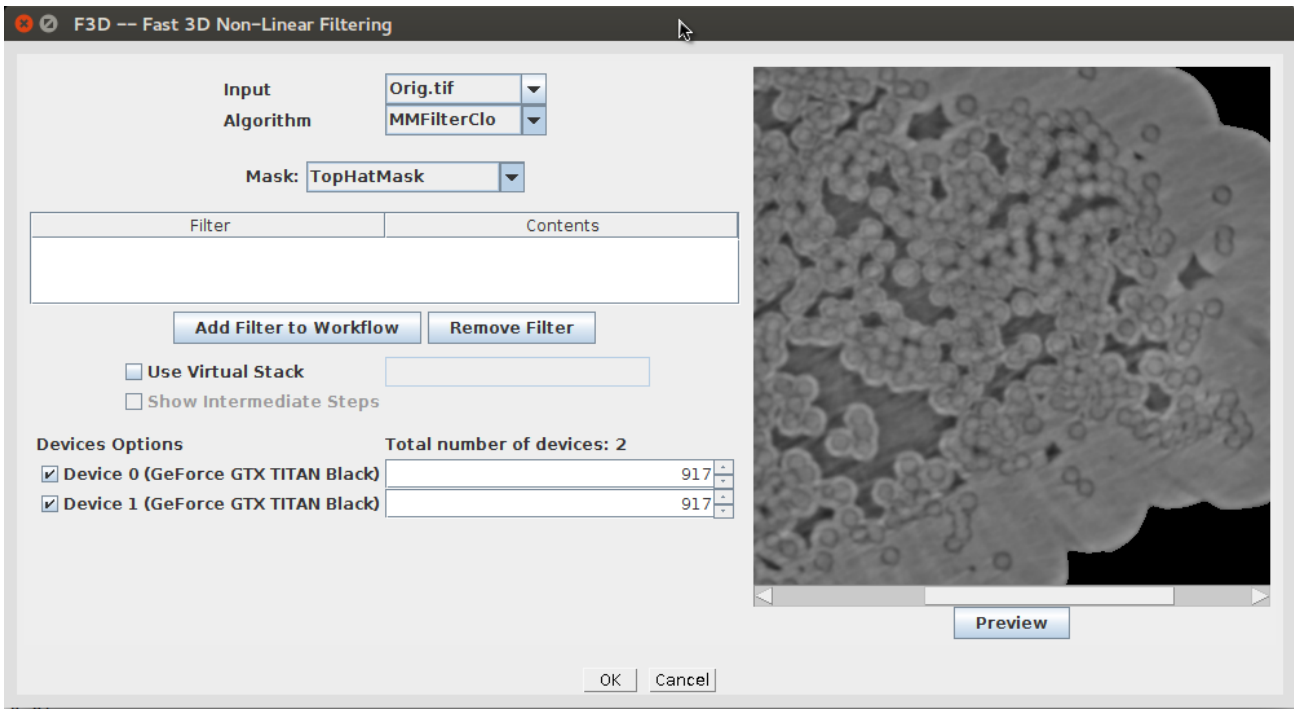


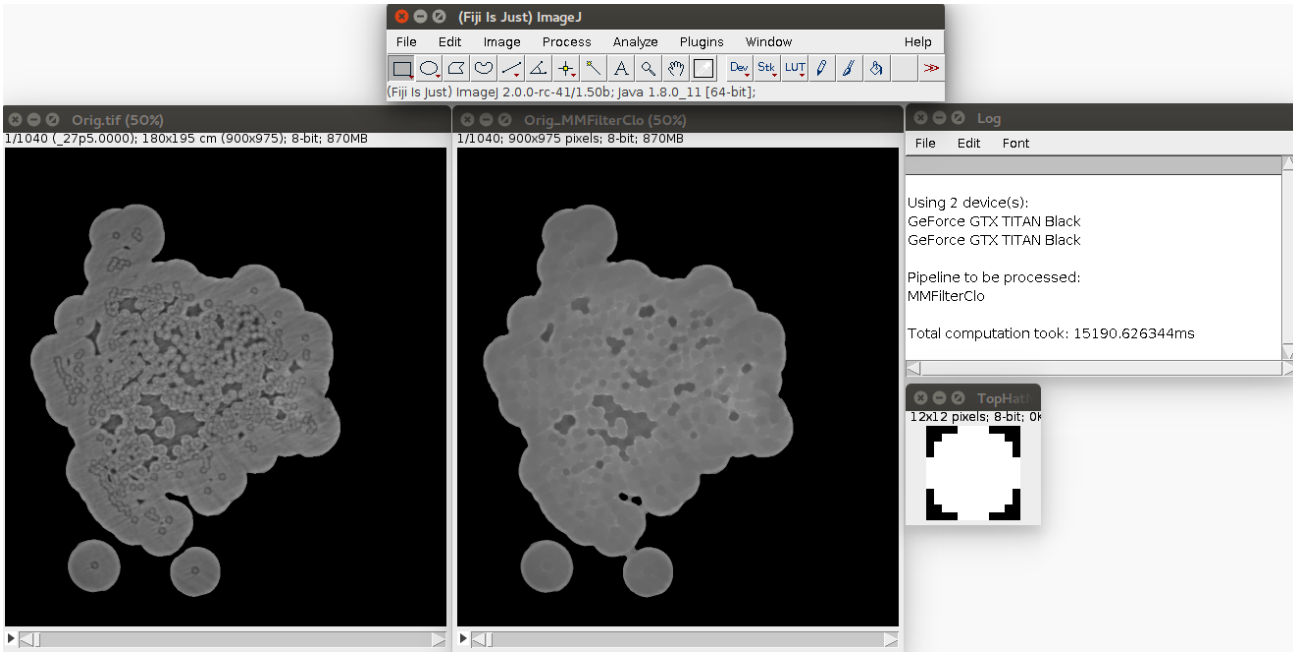

Orig.tif (50%)
1/1040 (_27p5.0000); 180x195 cm (900x975); 8-bit; 870MB
(Fiji Is Just) ImageJ
File Edit Image Process Analyze Plugins Window Help
x=177.60, y=43.20, z=0, value=0
New Image...
Name: TopHatMask
Type: 8-bit
Fill with: White
Width: 12 pixels
Height: 12 pixels
Slices: 1
OK Cancel


Orig.tif (50%)
1/1040 (_27p5.0000); 180x195 cm (900x975); 8-bit; 870MB
(Fiji Is Just) ImageJ
File Edit Image Process Analyze Plugins Window Help
(Fiji Is Just) ImageJ 2.0.0-rc-41/1.50b; Java 1.8.0_11 [64-bit];
TopHatMa
12x12 pixels; 8-bit; 0K

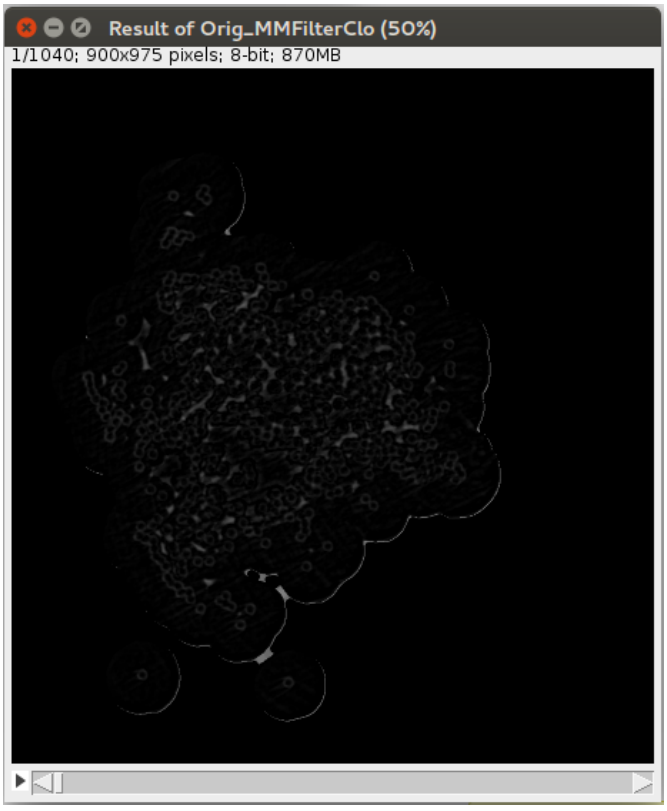


3. **Run closing operation using custom mask as structure element:**

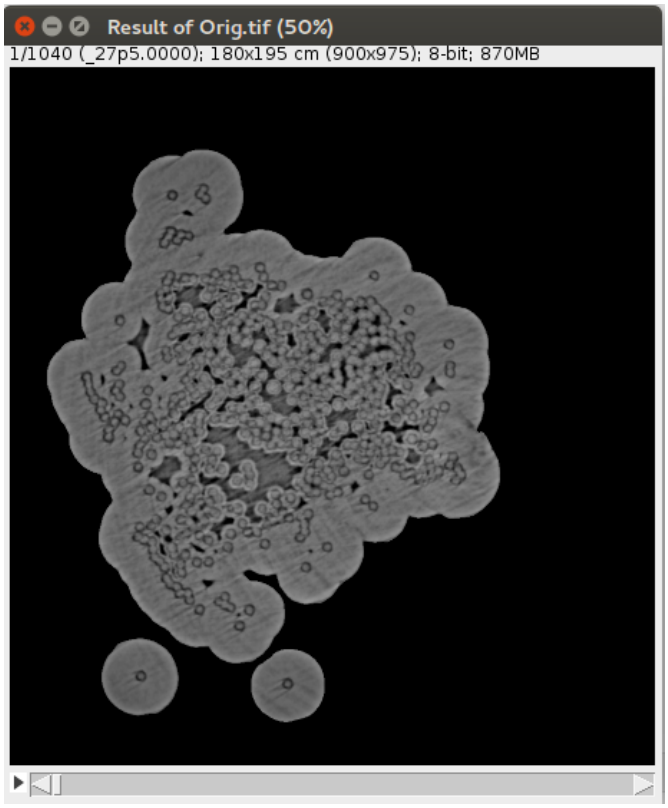




4. **Subtract 'Orig.tif' from 'Orig_MMFilterClo'**:



5. **Subtract 'Result of Orig_MMFilterClo' from 'Orig.tif':**



Running as an ImageJ macro

Improve contrast

```
// Macro to improve contrast using a custom structure element
open("Orig.tif");
newImage("TopHatMask", "8-bit white", 12, 12, 1);
makeOval(0,0,12,12);    setBackgroundColor(0,0,0);
run("Clear Outside","stack"); run("Select None");

// Change the name of input file accordingly
// MaxNumSlices should be less or equal to the number of slices of the
input image. To make sure about this number, before running the script,
open the input image and run F3D using the interface. This number is shown
next to the name of the device being used. Change the script accordingly
run("F3D Image Processing (JOCL)", "options=[{input : Orig.tif ,
intermediateSteps : false , devices : [{ Device: 0 , MaxNumSlices: 1040 } {
Device: 1 , MaxNumSlices: 1040 } ] , useVirtualStack : false ,
virtualDirectory : empty , filters : [{ Name : MMFilterClo , Mask :
[{maskImage : TopHatMask}] }]}]");
// Change the names of the images accordingly
imageCalculator("Subtract create stack", "Orig_MMFilterClo","Orig.tif");
imageCalculator("Subtract create stack", "Orig.tif","Result of
Orig_MMFilterClo");

selectWindow("Result of Orig_MMFilterClo");
close();
selectWindow("Orig_MMFilterClo");
close();
```