

---

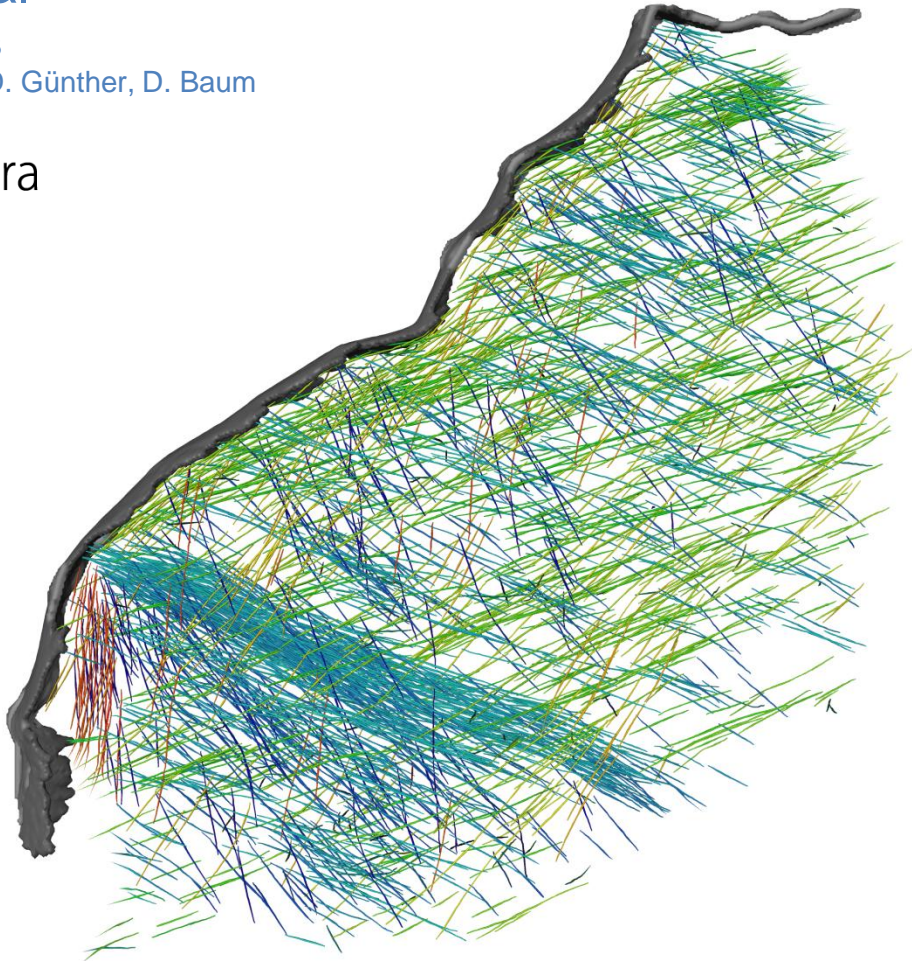
# AUTOMATED SEGMENTATION OF ACTIN NETWORKS USING THE VISUALIZATION FRAMEWORK AMIRA

## Tutorial

Version 1.3

A. Rigort, D. Günther, D. Baum

zibamira



KONRAD-ZUSE-ZENTRUM  
FÜR INFORMATIONSTECHNIK  
BERLIN

MAX-PLANCK-INSTITUT  
FÜR BIOCHEMIE  
MÜNCHEN



# zibamira

## Segmentation Tutorial

### INTRODUCTION

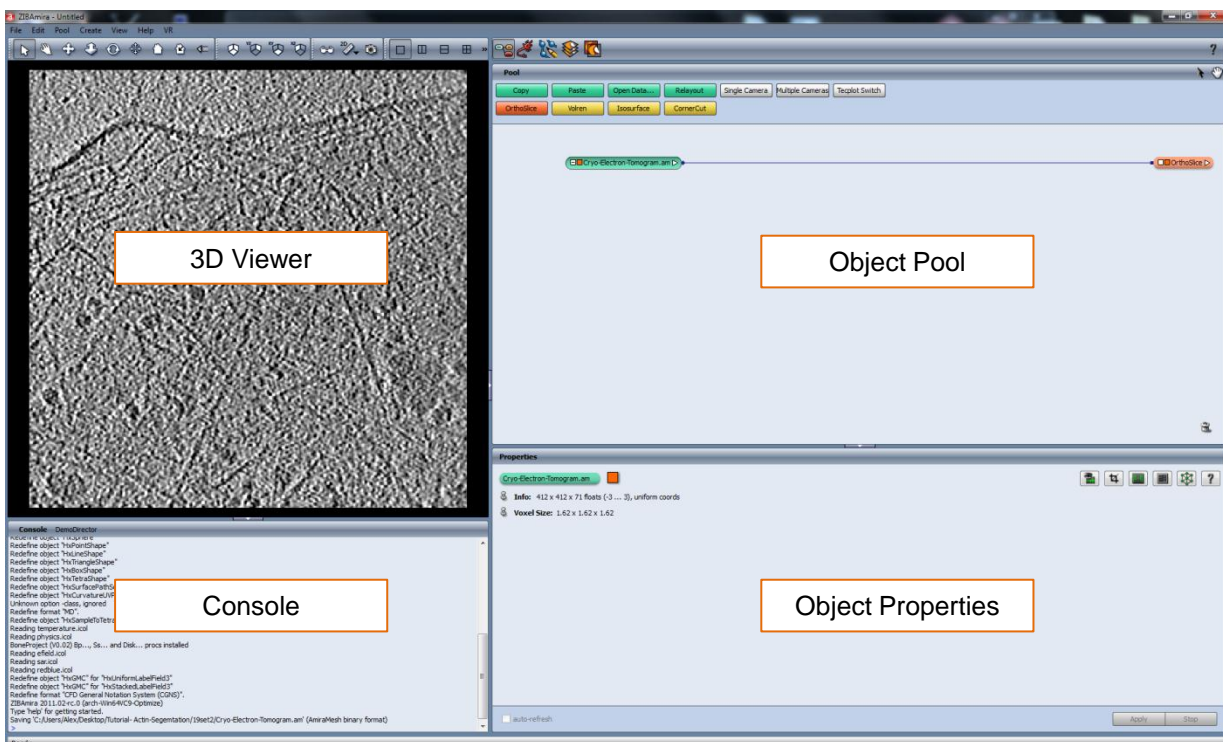
This tutorial is intended to walk you through the important steps required to extract **actin filaments** from cryo-electron tomograms using the actin segmentation package within AMIRA. After completing this tutorial, you should be able to extract actin filaments from your tomographic dataset. A basic introduction to AMIRA itself can be downloaded from:


[www.amira.com/images/stories/pdf/522/AmiraUsersGuide.pdf](http://www.amira.com/images/stories/pdf/522/AmiraUsersGuide.pdf).

The parameter settings used in this tutorial are adapted to tomographic data from the following type: 2048x2048 EM images (Voxel Size: 0.81 nm), Weighted Back Projection Reconstruction with Binning 1 (region shown: 412x412x70, Voxel Size: 1.62 nm , Datatype: **float**). Note: the y-axis of our 3D reconstruction corresponds to the tomography tiltaxis!

Workstation/Machine setup: DELL Precision T5400, Intel Xeon CPU E5440@2.83GHz dual quadcore processors, 64bit WIN7, 26GB RAM, GPU: Nvidia Quadro FX 5600.

### WORKSPACE LAYOUT IN AMIRA



For every module a help button  exists in the object properties viewport, where additional information can be found. The results of each computation step can be saved via the main menu *File > Save Data as...*

## SELECTING THE APPROPRIATE TEMPLATE FOR EXTRACTING CYTOSKELETAL FILAMENTS

The **shape** of the cylindrical template needs to fit locally the actin filaments. The width of the cylinder is therefore fixed by the diameter of **actin filaments**, which is approximately 7 nm. The length of the template is, however, more critical. Noise and incomplete information constrain the length. If the template is too short, it reacts to all fluctuations, resulting in arbitrary orientations. On the other hand, if the template is too long, it could miss small-scale features. Empirically, a template length of six times the width of an actin filament, i.e. 42 nm, turned out to be a good compromise and to yield satisfactory results. Smaller template lengths can be used, but the reliability of the extracted filaments becomes strongly dependent on the quality of the tomogram.

Similar to actin filaments, a solid cylindrical template can be used to extract **intermediate filaments**. Since these filaments have an approximate width of 14 nm the radius of the template should be set to 7 nm. For a good separation of these filaments from other structures a template length of 70 nm should be given.

**Microtubules** can be extracted using a hollow cylindrical template instead of a solid cylinder, as used for actin filaments. Two radii – the inner and the outer radius – must be specified by the user to define this template. The inner radius defines the size of the hollow space in the template while the outer radius defines the width of the microtubule. The approximate width of a microtubule is 25 nm. A value of 13-14 nm for the outer radius is therefore appropriate. The inner radius can be usually set to 7 nm. A template length of 120 nm yields reliable results such that microtubules can be separated from other structures in the tomogram.

In case all three cytoskeletal filaments are present within a tomographic dataset, the tracing approach is performed in a sequential manner: first for microtubules, then for intermediate filaments and finally for actin. Filamentous structures that have been identified in the previous steps are excluded from the subsequent tracing steps. This can be done by voxel-based masking using the line geometries found by the tracing algorithm.

## REMARKS ON PARAMETER SETTINGS

(using the *Extract Lines* Module, see page 7 f.)

- The **similarity** parameter (user defined threshold  $t_1$ ) influences the length of filaments traced. The higher this setting, the shorter the filaments.
- The **Min.Correlation** parameter (user defined threshold  $t_2$ ) influences the number of filaments traced. The higher this value, the lower the number of filaments.
- The **cocircularity** condition evaluates the smoothness of the filament. The user can favor straighter lines by choosing a small parameter, while large values allow more curved lines.
- The **linearity** function is the most sensitive parameter. It defines the opening angle of the search cone. Lowering the linearity leads to more, but shorter filaments.
- The **distance** function influences the length of the filaments traced. It defines the radius of the search cone.

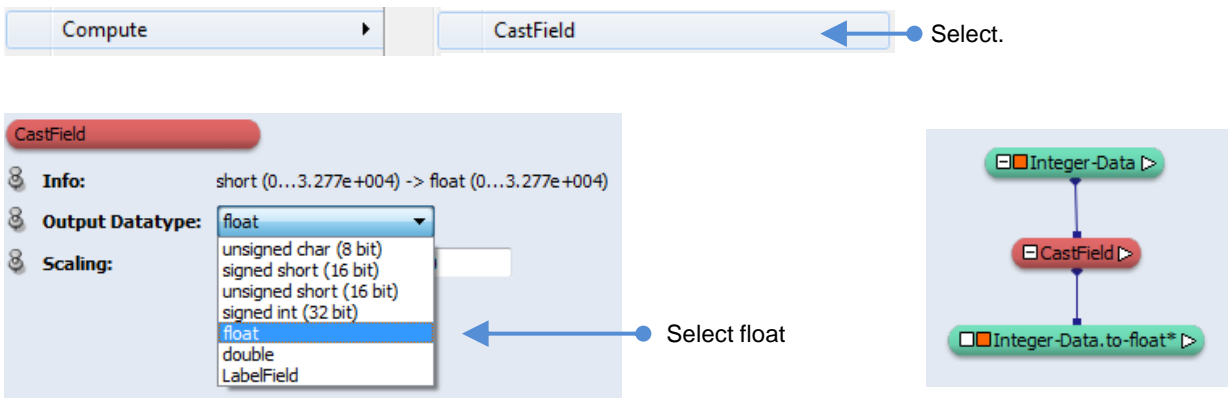
The parameters cocircularity, linearity and distance depend on the structure that is to be extracted. Those parameters must be adjusted once to the desired structure and can be kept throughout a series of experiments. For the actin filament data shown in this tutorial we have chosen the following configuration: cocircularity:  $0.05 \pi$ , linearity:  $0.3 \pi$ , distance: 40 nm. The parameters  $t_1$  and  $t_2$  have to be individually adjusted for each tomogram.

### CONVERSION OF DATA TYPES

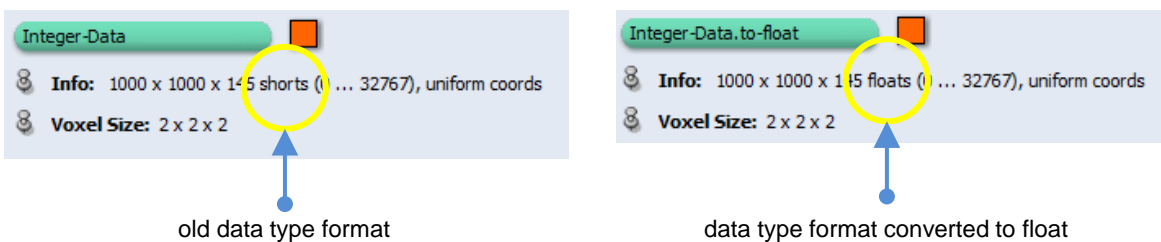
The input data type of the tomographic data used in this tutorial and in the *demo network* on the website <http://www.zib.de/en/visual/software/CryoEM.html> is **float**.

Depending on the parameters set in the reconstruction software, the data type of the 3D data set can be different. In IMOD, e.g., the data type/storage mode of the output file can be set to: 0 for byte, 1 for 16-bit signed integer, 6 for 16-bit unsigned integer or 2 for 32-bit floating point.

If you want to convert your data set to float data you can easily do this in AMIRA by using the *CastField* module. Simply right click on your data file and connect *Compute > CastField* to the data set and select *Output Datatype: float*.



In the object properties viewport hit the **Apply** button to start the conversion  
You can check the data type by clicking on the data object. The data type is displayed in the *Info*: line



# zibamira

## Segmentation Tutorial

### WORKFLOW STEPS SHOWN IN THIS TUTORIAL

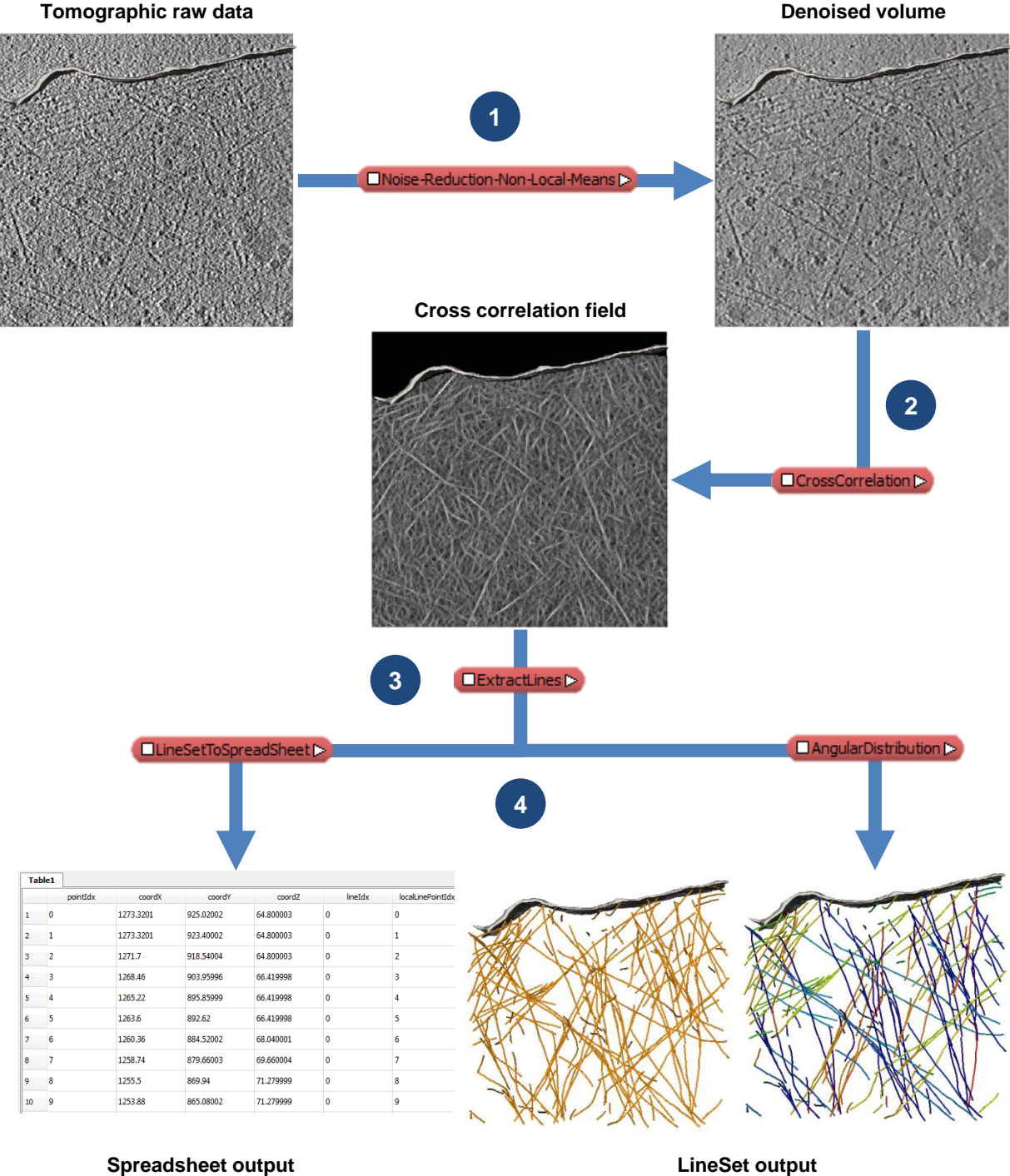
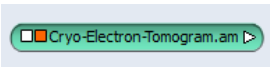


Table1						
	pointIdx	coordX	coordY	coordZ	lineIdx	localLinePointIdx
1	0	1273.3201	925.02002	64.800003	0	0
2	1	1273.3201	923.40002	64.800003	0	1
3	2	1271.7	918.54004	64.800003	0	2
4	3	1268.46	903.95996	66.419998	0	3
5	4	1265.22	895.85999	66.419998	0	4
6	5	1263.6	892.62	66.419998	0	5
7	6	1260.36	884.52002	68.040001	0	6
8	7	1258.74	879.66003	69.660004	0	7
9	8	1255.5	869.94	71.279999	0	8
10	9	1253.88	865.08002	71.279999	0	9

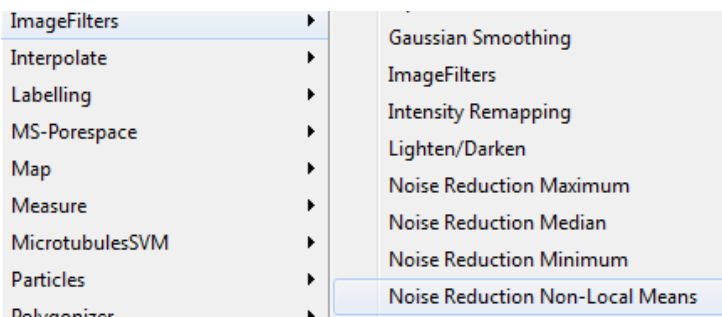
### 1

## DENOISING THE RAW DATA CRYO-ELECTRON TOMOGRAM

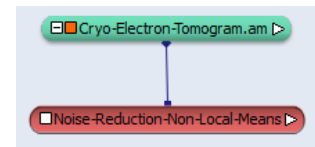
Load your tomographic volume (\*.em, \*.mrc) via *File > Open Data ...* or drag&drop your file directly into the object pool.



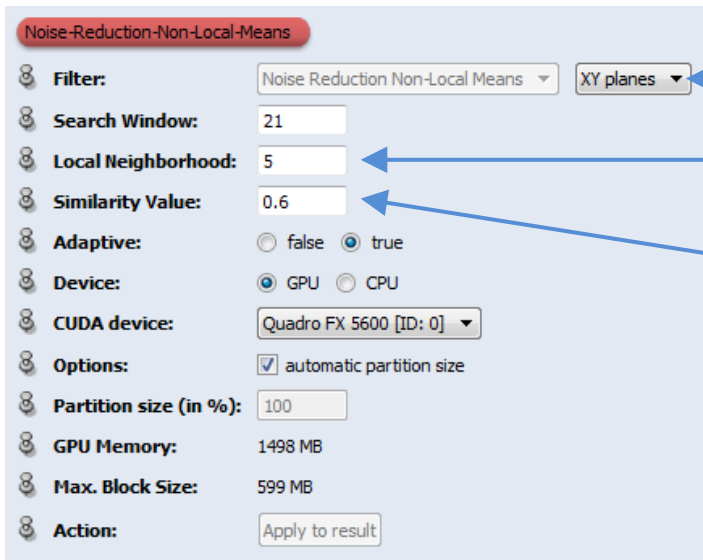
Right click on the file and select the denoising filter via *ImageFilters > Noise Reduction Non-Local Means*:



connected module:



In the AMIRA object properties viewport enter the following parameters:



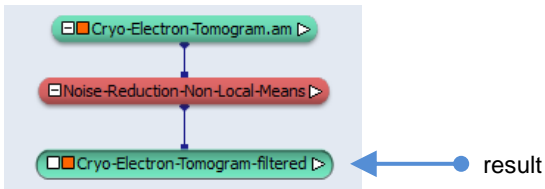
Operation in 2D or 3D can be selected. 3D is computationally more expensive.

Radius of the neighborhood window in voxels/pixels.

The bigger this weighting factor is, the smoother the result will be.

In the object properties viewport hit the  button to start computation.

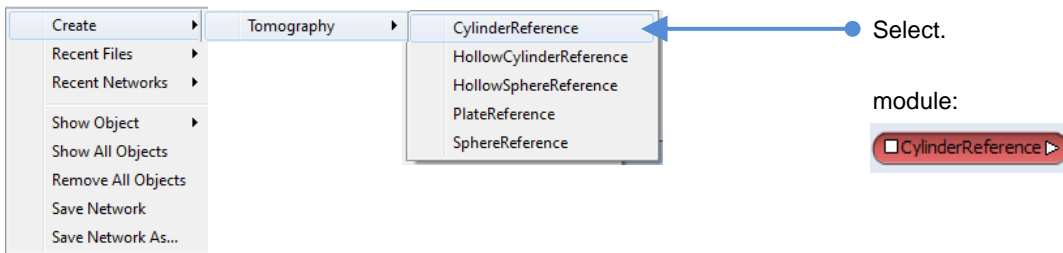
The result of the denoising step is shown as a new object in the object pool.



## 2

### SELECTION AND ADJUSTMENT OF THE CYLINDRICAL TEMPLATE

In the object pool right click in the background and select: *Create > Tomography > CylinderReference*



Enter the following parameters in the object properties viewport:

Length of the template in nanometers (unit depends on the dimension of the input data).

This value must correspond to the voxel size of the tomogram. It is usually given in nm.

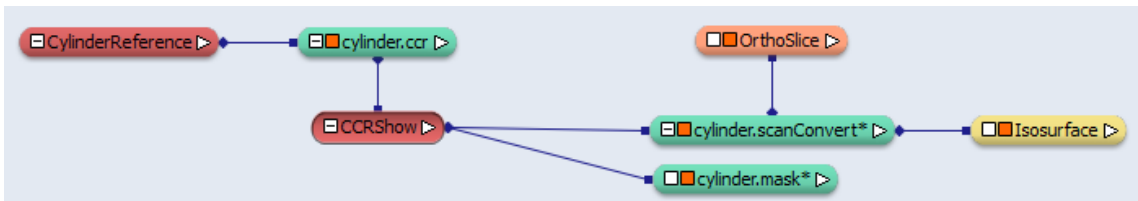
Allows the user to smooth the template. If the value is 0 no smoothing is done at all.

Defines the missing wedge of the tomogram.

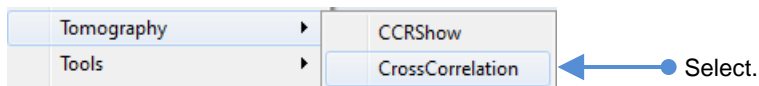
Hit the  button for additional information about the module properties.

In the object properties viewport hit the  button to start computation.

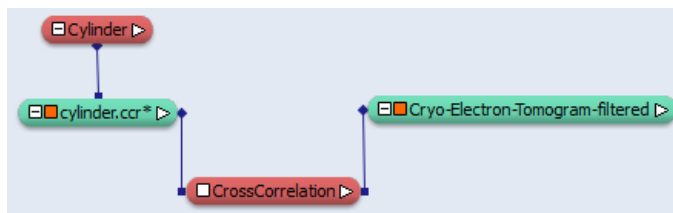
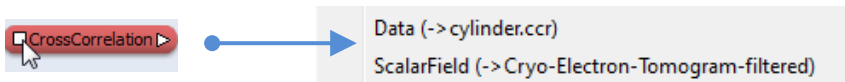
The result is shown in the object pool (*cylinder.ccr\**). For a visualization of the generated template the following network can be constructed: Right click on the *cylinder.ccr* object and select *Tomography > CCRShow*. In the *CCRShow* module click the auto-refresh clickbox. Two new objects appear. Connect an *Isosurface* to the *cylinder.scanConvert* object and set it to auto-refresh by clicking the checkbox. The different template orientations can be visualized by moving the index slider in the *CCRShow* module.



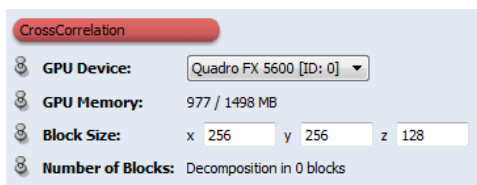
To select the cross correlation module right click at the *cylinder.ccr* object and select: *Tomography > CrossCorrelation*



Connect the denoised tomographic volume to the *CrossCorrelation* module by clicking with the mouse into the small white square of the *CrossCorrelation* module, select *ScalarField* and click on the denoised tomographic volume.

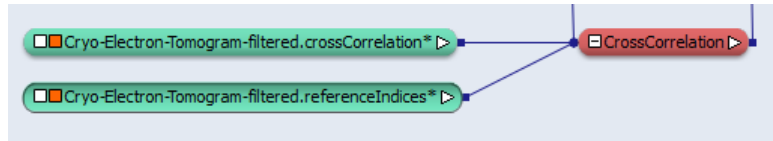


The cross correlation is computed on the GPU. Parameters for the computation can be set in the object properties viewport. Note: if an error message pops up (due to a GFX card hardware conflict), it is recommended to reduce the values for *Block Size*.



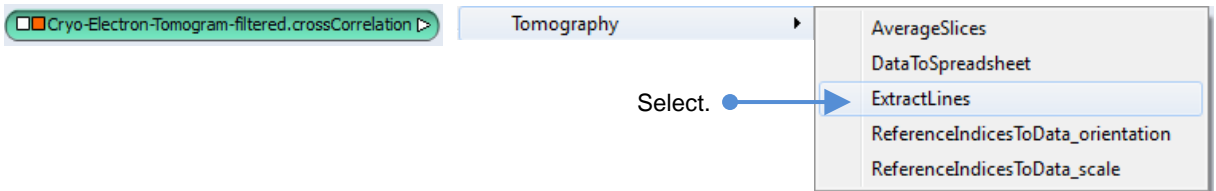
In the object properties viewport hit the Apply button to start computation.

The output from the *CrossCorrelation* module are two new data objects, the correlation field (termed *crossCorrelation*) and orientation field (termed *referenceIndices*). Both files can be saved via *File > Save Data as...*

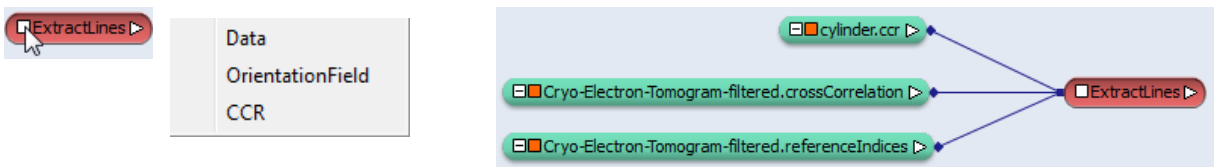


### 3 TRACING OF FILAMENTS

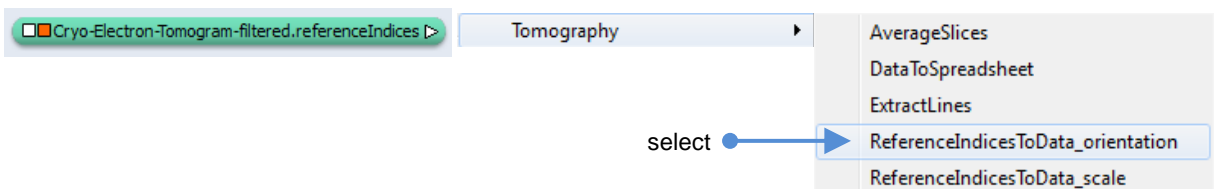
In order to start the filament tracing procedure right click on the green *crossCorrelation* object and select *Tomography > ExtractLines*.



There are two options to extract the filaments: **Option A** is less memory consuming, but filament tracing takes longer. Click in the small white box of the generated *ExtractLines* module and connect the *cylinder.ccr* object via the CCR port and the *referenceIndices* object via the OrientationField port.

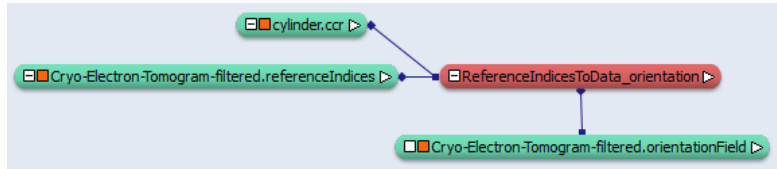


**Option B** converts the information from the *referenceIndices* object into an *orientationField* object. This variant is consuming more memory, but filament tracing is much faster. Right click on the green *referenceIndices* object and select *Tomography > ReferenceIndicesToData\_orientation*.

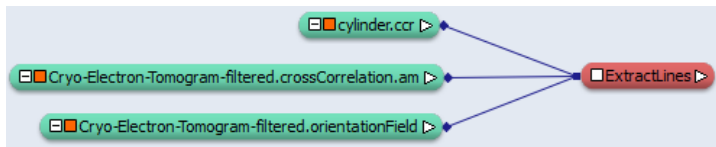


In the object properties viewport hit the  button to start computation.

The result is a new object called *orientationField*.



Now, create an *ExtractLines* module in the same way as shown for option A. Instead of attaching the *referenceIndices* object the new *orientationField* object is connected. The other two objects (*cylinder.ccr* and *crossCorrelation*) are connected accordingly.



Once the three data objects are connected to the *ExtractLines* module, the tracing parameters can be adjusted in the object properties viewport. The simple GUI allows the user to adjust the similarity value and cross correlation coefficient used for tracing:

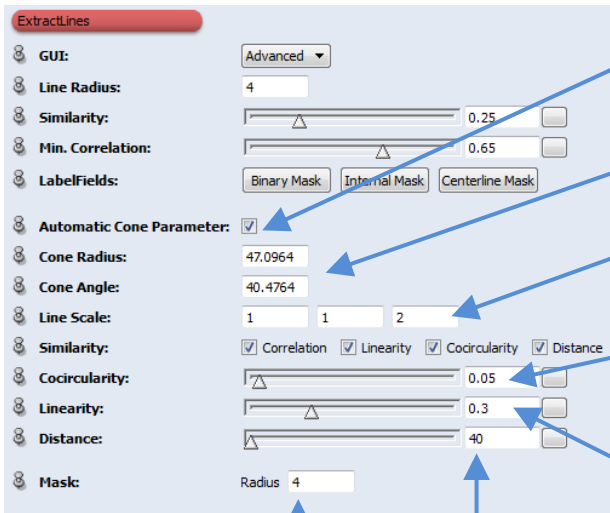
- Allows switching between simple or advanced menu.
- Defines the diameter of the filaments (here: 8 nm).
- Influences the length of filaments traced. (threshold  $t_1$ )
- Influences the number of filaments traced. (threshold  $t_2$ )
- Creates a voxel-based representation of the filament center lines (available after the tracing process).
- Creates a voxel-based representation of the filaments approximating elongation effects caused by the missing wedge (available after the tracing process).
- Creates a voxel-based representation of the traced filaments (available after the tracing process).

In the object properties viewport hit the  button to start computation.

# zibamira

## Segmentation Tutorial

The advanced GUI allows to adjust more parameters for filament tracing:



Automatic adjustment of the cone parameters. [recommended]

These values are calculated automatically if the checkbox above is ticked.

Takes into account the elongation in x,y,z.

Influences the smoothness of a filament. Small values yield straighter lines, while large values allow more curved lines.

Most sensitive parameter. Lowering this parameter leads to more, but shorter filaments. Defines the opening angle of the search cone.

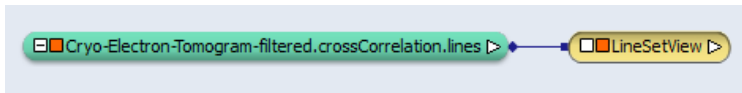
Influences the length of filaments traced. Defines the radius of the search cone.

This parameter is not of importance for tracing, but defines the radius of the binary mask.

In the object properties viewport hit the  button to start computation.

### 4 VISUALIZATION AND ANALYSIS OF THE TRACED FILAMENTS

The result of the line tracing can be visualized by connecting a *LineSetView* module via right clicking the green *crossCorrelation.lines* data object.

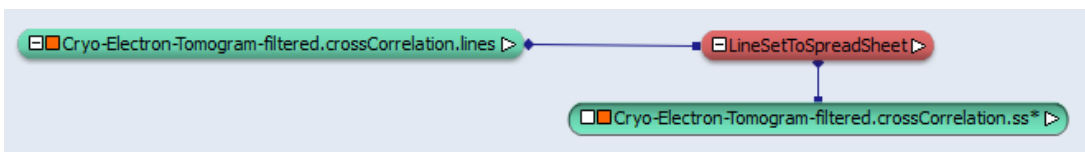
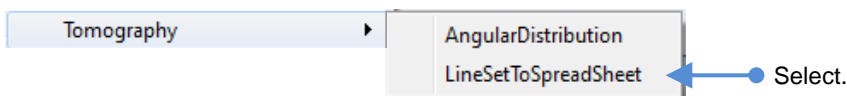


For a visualization of the traced filaments, the line settings can be adjusted in the *LineSetView* properties viewport:

The screenshot shows the 'LineSetView' properties panel with several settings and blue arrows pointing to them from explanatory text on the right:

- Shape:** Set to 'Circle'. Annotation: "Choosing *circle* allows scaling the lines according to the diameter of the filaments."
- Circle Complexity:** Set to 12.
- Scale Mode:** Set to 'Constant'. Annotation: "Choosing *Constant* selects a constant scaling of the lines. Choosing *Data 0* selects scaling with respect to the correlation values along the lines. Choosing *Data 1* selects scaling with respect to the computed similarity values along the lines."
- Scale Factor:** Set to 4. Annotation: "Scaling factor."
- Color Mode:** Set to 'Data 0'. Annotation: "Select *Data 0* or *Data 1* to enable line coloring."
- Colormap:** A color bar ranging from 0 to 1. Annotation: "Allows the user to set the color of filaments."

For **postprocessing** and **analysis** of the traced filaments outside of AMIRA, data representing geometry of individual filaments can be displayed and exported. In order to do this right click at the *crossCorrelation.lines* object and select *Tomography > LineSetToSpreadSheet*.



Hit the  button to generate the spreadsheet (\*.ss file)

In the *crossCorrelation.ss* object press the Show button to display a table with the individual line indices (lineIdx), corresponding point indices along a line (pointIdx), the local line point indices (localLinePointIdx) and coordinates of the traced filaments within the 3D volume. *Data 0* shows the cross correlation values and *Data 1* the corresponding scalar values for each point.

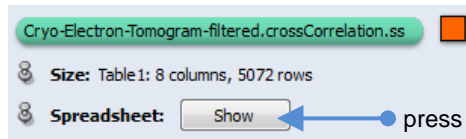
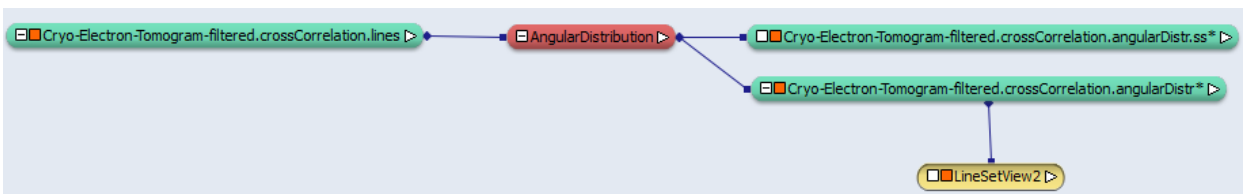


Table1								
	pointIdx	coordX	coordY	coordZ	lineIdx	localLinePointIdx	Data 0	Data 1
1	0	1273.3201	925.02002	64.800003	0	0	0.37573704	0.27550539
2	1	1273.3201	923.40002	64.800003	0	1	0.42490369	0.41128027
3	2	1271.7	918.54004	64.800003	0	2	0.49777353	0.40263379
4	3	1268.46	903.95996	66.419998	0	3	0.5511924	0.50490344
5	4	1265.22	895.85999	66.419998	0	4	0.52987003	0.476403
6	5	1263.6	892.62	66.419998	0	5	0.48010305	0.43085015
7	6	1260.36	884.52002	68.040001	0	6	0.48179018	0.45828602
8	7	1258.74	879.66003	69.660004	0	7	0.4819895	0.39470428
9	8	1255.5	869.94	71.279999	0	8	0.53835642	0.52782178
10	9	1253.88	865.08002	71.279999	0	9	0.54393923	0.4262071

Spreadsheet containing geometry information about each traced filament

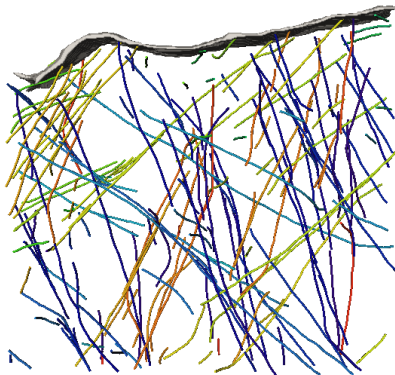
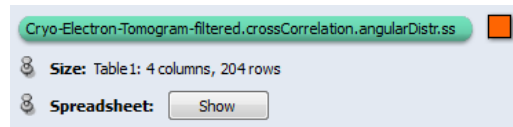
To display the angular distribution of the filaments, right click at the *crossCorrelation.lines* object and select *Tomography > AngularDistribution*.

Hit the **Apply** button to generate the spreadsheet (\*.ss file) and a new line object (*angularDistr*) which can be visualized by another LineSetView module.



This angular distribution module estimates the angular orientation of each line. The estimation is done in a least square sense. A linear function is fitted such that the distance of each line point is minimized.

The output is a spreadsheet (\*.ss file) containing the estimated angles phi and theta. Phi is the angle in the x,y-plane with respect to the y-axis in counter-clockwise direction. Phi is in the range of [0, 180] degrees. Theta describes the elevation of the line with respect to the x,y-plane. Theta is in the range of [-90, 90] degrees. Additionally, the euclidian length of each line is stored in the spreadsheet.

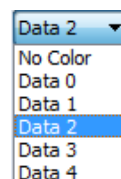
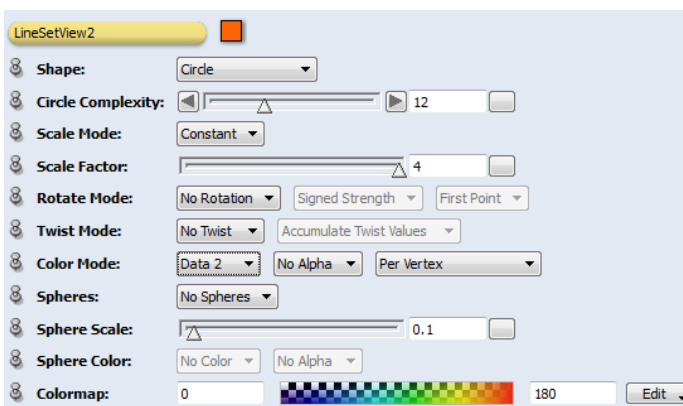
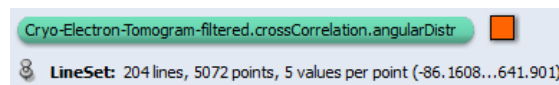


LineSetView of the traced filaments

Table1				
	lineIdx	phi	theta	lineLength
1	0	20.217533	-1.6439694	301.94354
2	1	143.98438	-2.6886809	382.51645
3	2	153.63138	-3.9076443	244.98022
4	3	143.95248	-8.1724148	245.58459
5	4	116.56505	2.5044781e-006	3.6224191
6	5	128.72113	1.3235908	428.7146
7	6	138.94682	-5.1416955	242.15598
8	7	150.32333	-1.2816347	238.57405

Spreadsheet

The second output is a new set of lines (*angularDistr*) containing three additional data channels (Data 2-4). Data 2 includes the angle phi, Data 3 the angle theta and Data 4 the euclidian line length.

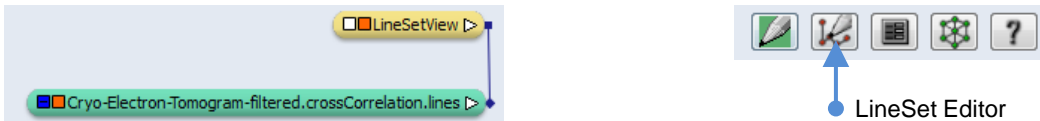


- Data 0 = cross correlation
- Data 1 = similarity value
- **Data 2 = phi value (x,y plane)**
- Data 3 = theta value
- Data 4 = euclidian length

The *LineSetView* module can be used to visualize these data channels accordingly (for adjusting the colormap right click in the color field, select e.g. *physics.col* colormap and use the *Adjust to data range* function).

**Note:** The cell membrane (see workflow diagram on page 4) or internal membranes (vesicles, organelles) should be masked out after the *CrossCorrelation* procedure and before the subsequent filament *tracing* step, since a cylindrical template yields high correlation values at these structures. In addition, false traced lines/filaments can be removed by visual inspection using the *LineSet Editor*.

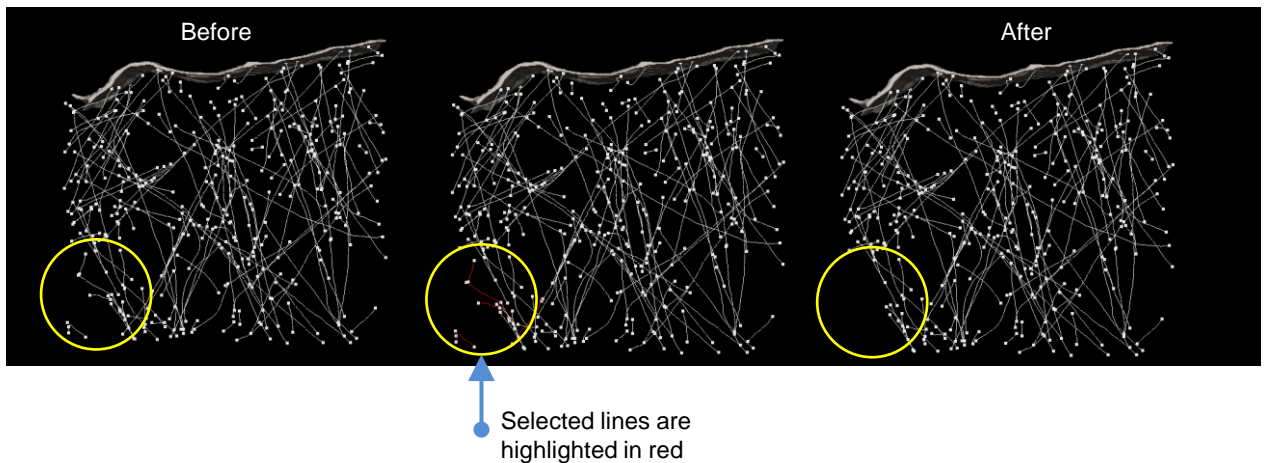
Click on the green *crossCorrelation.lines* data object and select the *LineSet Editor* from the object properties viewport.



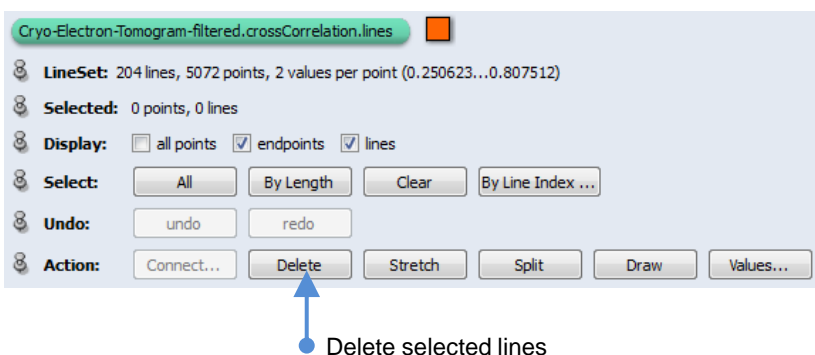
Select the interact pointer from the 3D viewer tools:



Now click in the 3D viewer on the lines/filaments you want to remove (hold down SHIFT to select multiple lines):



Press the *Delete* button in the *crossCorrelation.lines* object properties to delete the selected lines:



## CORRESPONDING PUBLICATION

Additional information about automated segmentation of electron tomograms can be found in the Journal of Structural Biology:

***Automated segmentation of electron tomograms for a quantitative description of actin filament networks***,  
A. Rigort, D. Günther, R. Hegerl, D. Baum, B. Weber, S. Prohaska, O. Medalia, W. Baumeister, H.C. Hege;  
Journal of Structural Biology: [doi:10.1016/j.jsb.2011.08.012](https://doi.org/10.1016/j.jsb.2011.08.012)

## ROOM FOR NOTES