Filopodia Tool User Manual

The semi-automatic workflow and a dedicated set of tools have been developed within the *Filament Editor* (FE) framework of AmiraZIBEdition. The FE provides 2D/3D visualization and interactive editing of filamentous, graph-like objects embedded in 3D space (hence called SpatialGraphs in Amira). The 2D view can display an image cross-section with the graph superimposed and offers tools for interactive tracing. The 3D view displays the graph and optionally a 3D direct volume rendering visualization of the image data. The FE was extended by an additional toolbox called *Filopodia* to be able to process time-dependent geometry and support the functionality described below. This toolbox is subdivided into three parts related to the subtasks: visualization, growth cone tracing and preprocessing and filopodia tracing and tracking.

In the early versions of the software, the term *growth cone* was used, but later with generalization and use of software on different cell types, the term *axon terminal* became preferential. In software and manual, they are used interchangeably.

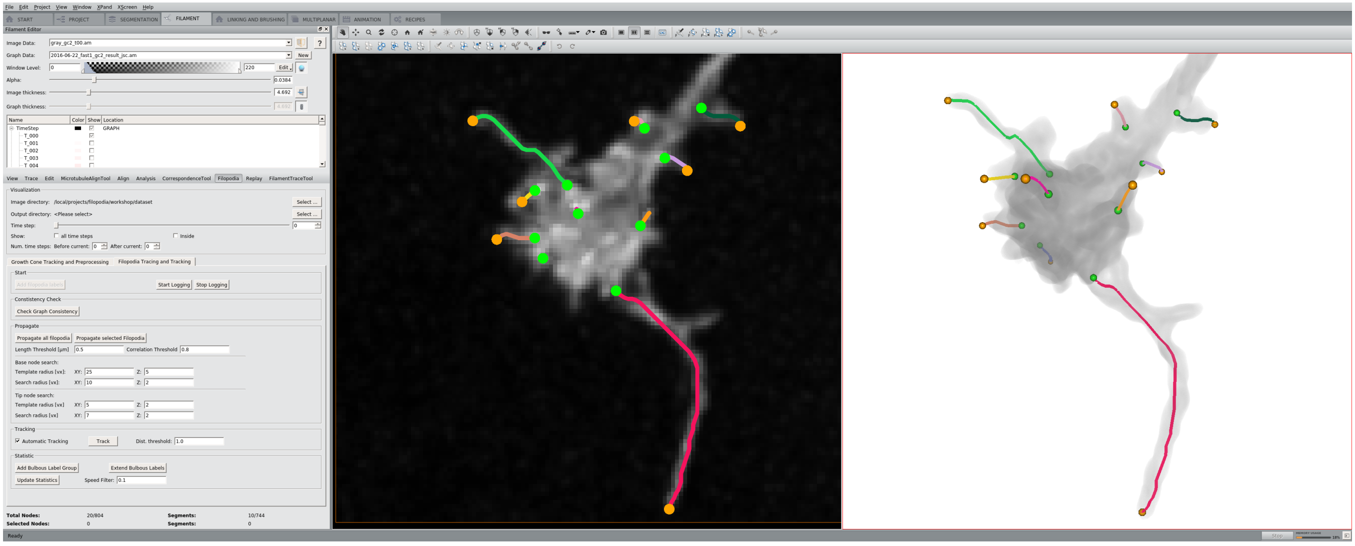


Image Data: gray_gc2_t00.am

Graph Data: 2016-06-22_fast1_gc2_result_jsc.am

Window Level: 0 220

Alpha: 0.0384

Image thickness: 4.692

Graph thickness: 4.692

Name	Color	Show	Location
TimeStep	■	☑	GRAPH
T_000		☑	
T_001		□	
T_002		□	
T_003		□	
T_004		□	

Visualization

Image directory: /local/projects/filopodia/workshop/dataset

Output directory: <Please select>

Time step: 0

Show: □ all time steps □ Inside

Num. time steps: Before current: 0 After current: 0

Start

Consistency Check

Propagate

Length Threshold [μm] 0.5 Correlation Threshold 0.8

Base node search:

Template radius [vx]: XY: 25 Z: 5

Search radius [vx]: XY: 10 Z: 2

Tip node search:

Template radius [vx] XY: 5 Z: 2

Search radius [vx] XY: 7 Z: 2

Tracking

☑ Automatic Tracking Dist. threshold: 1.0

Statistic

Speed Filter: 0.1

Total Nodes: 20/804 Segments: 10/744

Selected Nodes: 0 Segments: 0

Figure 1: The FE offers a 2D and 3D viewer. The *Filopodia Toolbox* offers several tools for the semi-automatic workflow.

Visualization

For visualization, the *Filopodia* toolbox offers options to load data and move through time steps using a slider.

pressing the button *Select* next to *Image directory* opens a data explorer where the user can choose the dataset to be worked on. Depending on the part of the workflow, the necessary input varies:

- Growth cone tracing and preprocessing
  - If one wants to start the semi-automatic workflow with preprocessing, the folder containing the 3D grayscale images of the dataset has to be selected. Usually, there are 60 am-files (one file per time step). The filenames should end with `*_T<x>.am` or `*_t<x>.am`, whereby *x* is the time step.
  - Example: `*/<experiment-folder>/am/`
- Filopodia tracing and tracking

- If one wants to reconstruct filopodia, the folder containing the data of the axon terminal needs to be selected. In this folder, a spatial graph `rootNodes_gc<n>.am` representing the axon terminal centers for each time step and the subfolders `Gray` and `Dijkstra` are stored. `Gray` contains 60 am-files of cropped grayscale images showing the axon terminal. `Dijkstra` contains 60 am-files storing the precomputed dijkstra maps required for tracing.
- Example: `*/<experiment-folder>/results/GrowthCone_<n>/`. For the preprocessing, one needs to set the output directory, where the results will be stored. This can be achieved by pressing *Select* next to *Output directory* and choosing the directory using the opened data explorer. The explorer also offers to add new folders that can be selected afterward, too.

To process the time-dependent geometry of the filopodia graphs, the *Filopodia* toolbox offers the *Time slider*, where one can set the wanted time step to observe. While changing the time step, the grayscale image in 2D and 3D viewer is updated to the new time step.

If option *all time steps* is enabled, the entire filopodia graph is shown. If not, only the graph components with the current time step label ID are presented. One can visualize prior and/or succeeding time steps by increasing the *Before current* or *After current* window.

The checkbox *only filopodia* offers to show only the part between tips and base nodes. Those components are labeled as *filopodium* (read Location for more information). This option can make visualization clearer since those filopodia parts are relevant for filopodia geometrical reconstruction.

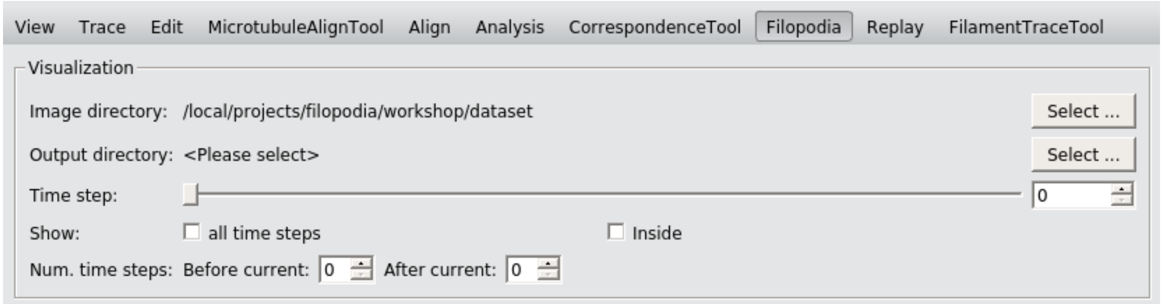


Image directory: /local/projects/filopodia/workshop/dataset

Output directory: <Please select>

Time step: 0

Show: □ all time steps □ Inside

Num. time steps: Before current: 0 After current: 0

Figure 2: The visualization part of the filopodia toolbox provides loading images and Dijkstra maps for the semi-automatic workflow. Additionally, a time-slider allows stepping through time. The user also has options to enable or disable several parts of the filopodia graph (show all time steps/show only filopodia). Furthermore, the user can display prior or succeeding parts (time steps) of the graph.

Growth cone tracing and preprocessing

One has to load the images of the dataset and select an output directory (see visualization).

In the beginning, the user needs to set the axon terminal centers in the first time step and activate the root node propagation by clicking *Propagate root nodes*. The radii of the required template and search window can be set manually. Due to the anisotropic voxel size, the search window can differ in Z dimension.

To reduce computation time, data size and to avoid occlusions, the axon terminals need to be separated. This is achieved by cropping the original image and saving each axon terminal as a separate image. Therefore, a bounding box around each axon terminal is computed. The size of the bounding box of an axon terminal is identical in all the time steps, but the box is shifted relative to the axon terminal center. The user can display the bounding box of any axon terminal by enabling the checkbox *show box in 3D viewer*.

An estimation of box location and size can be automatically computed by clicking the button *Auto estimate bounding box*. It is based on the Contour Tree Segmentation, which requires an intensity threshold as well as a persistence value. Those values are user-adjustable and can be set in the *Filopodia* toolbox. Automatic estimation can be unreliable for datasets containing many axon terminals.

The user can also adjust the bounding box for each axon terminal manually by defining the box size in X, Y and Z and also the location relative to the axon terminal center (root node). The user has to make sure that the axon terminal fits in the box for all the time steps.

After defining suitable bounding boxes for all axon terminals, the user can press *Crop + Dijkstra* to start the cropping and Dijkstra maps computation. For the latter, one can also set *Intensity Weight* and *Top Brightness* parameters. For each axon terminal, the cropped grayscale images as well as the precomputed Dijkstra maps are stored in the selected output directory.

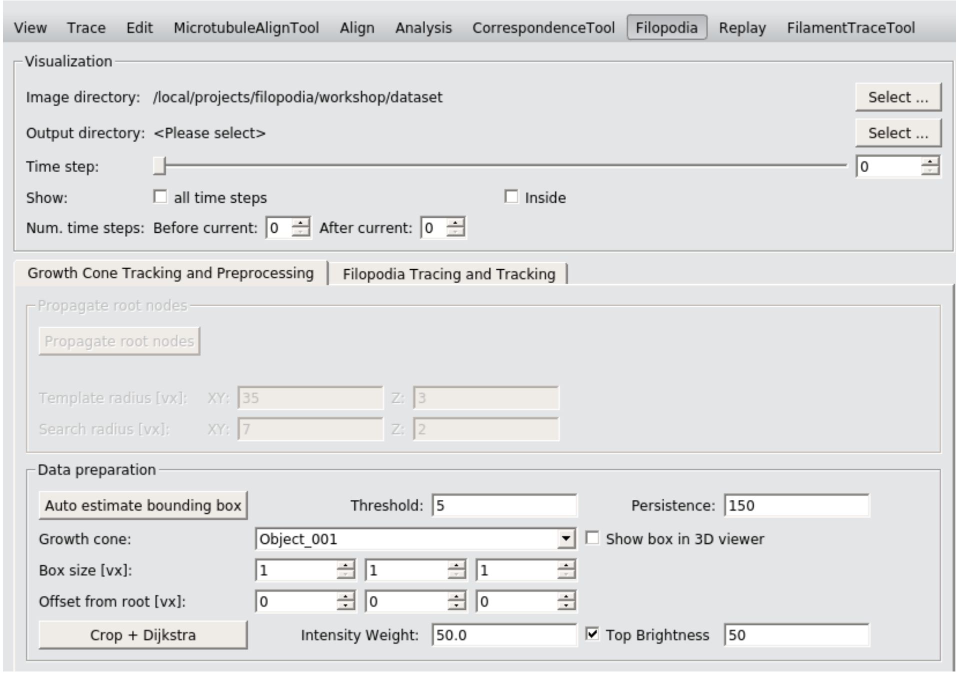


Image directory: /local/projects/filopodia/workshop/dataset

Output directory: <Please select>

Time step: 0

Show: □ all time steps □ Inside

Num. time steps: Before current: 0 After current: 0

Template radius [vx]: XY: 35 Z: 3

Search radius [vx]: XY: 7 Z: 2

Threshold: 5 Persistence: 150

Growth cone: Object_001 □ Show box in 3D viewer

Box size [vx]: 1 1 1

Offset from root [vx]: 0 0 0

Intensity Weight: 50.0 ☑ Top Brightness 50

Figure 3: The tabular *preprocessing* contains tools for root node propagation and data preparation. The latter includes the tool for image cropping, as well as Dijkstra maps pre-computation.

Filopodia Tracing and Tracking

The main part of the semi-automatic filopodia workflow can be done by the *Filopodia Tracing and Tracking* tab.

Before starting, the user has to add some filopodia labels (Location, Filopodia, ManualNodeMatch, and ManualGeometry) by pressing *Add filopodia labels*. Otherwise, the filopodia tools and other buttons are not available.

By pressing *Check Graph Consistency* one can update the consistency table at any time during the process.

After tracing filopodia in a single time step, the user can propagate filopodia into the following time step by pressing the button *Propagate all filopodia*. If one only wants to propagate one or more selected filopodia push *propagate selected filopodia*. Filopodia propagation needs the template and search window parameters for base and tip node propagation. Additionally, the user can define a minimum filopodia length and a correlation threshold.

Usually, the filopodia IDs will be updated after each spatial graph change. But for very large filopodia graphs with many filopodia, this tends to be very time consuming. Therefore, we added the possibility the disable the automatic tracking by unchecking the box *Automatic Tracking*. Instead, the user has to activate the tracking manually from time to time by pressing the button *Track*.

The filopodia statistics can be computed by pressing *Update Statistics*. The user can adjust the *Speed Filter* to define the threshold of static filopodia. Additionally, the optional bulbous label can be added or extended.

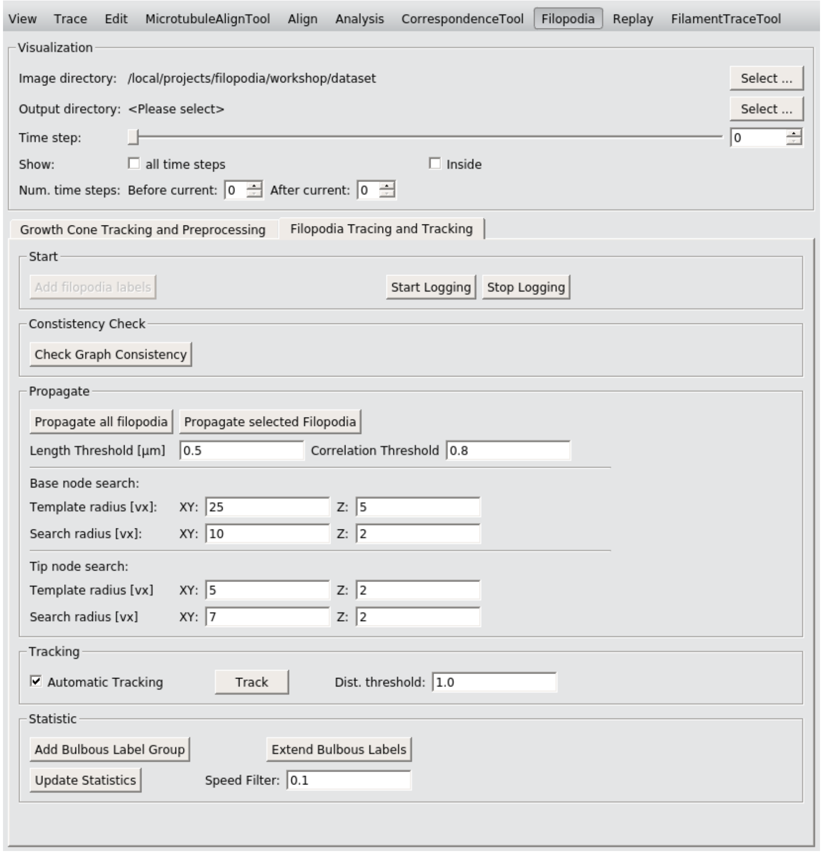


Image directory: /local/projects/filopodia/workshop/dataset

Output directory: <Please select>

Time step: 0

Show: □ all time steps □ Inside

Num. time steps: Before current: 0 After current: 0

Length Threshold [µm] 0.5 Correlation Threshold 0.8

Base node search:

Template radius [vx]: XY: 25 Z: 5

Search radius [vx]: XY: 10 Z: 2

Tip node search:

Template radius [vx] XY: 5 Z: 2

Search radius [vx] XY: 7 Z: 2

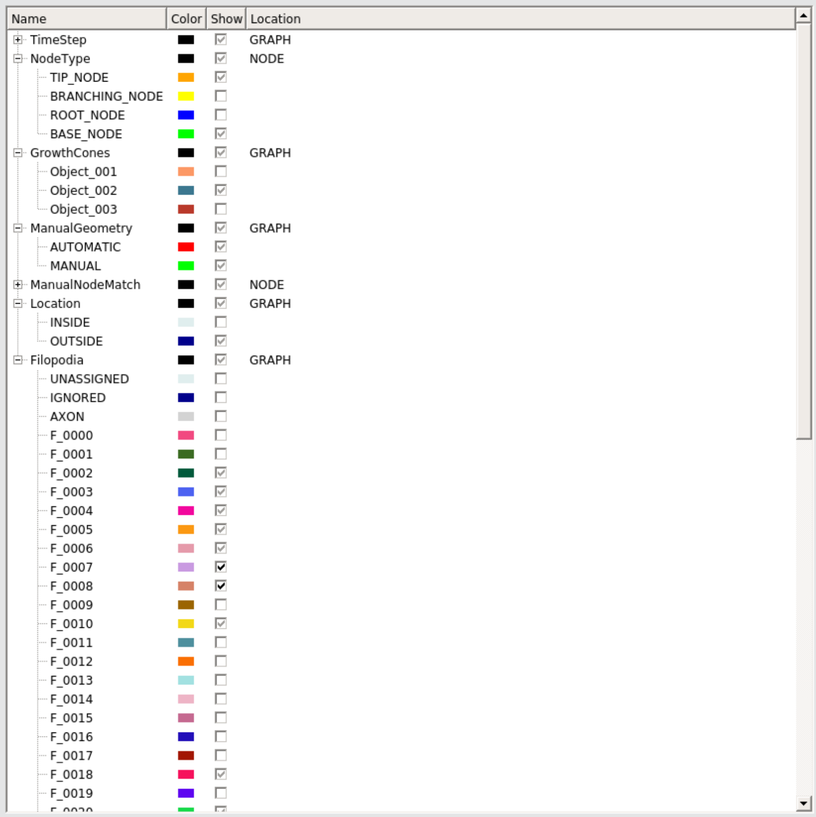
☑ Automatic Tracking Dist. threshold: 1.0

Speed Filter: 0.1

Figure 4: The tab *Filopodia Tracing and Tracking* offers several options required for the semi-automatic workflow: Filopodia propagation including base and tip propagation, tracking, as well as computing of statistics with or without bulbous filopodia.

Filopodia labels

The filopodia graph offers multiple labels to save more information like time steps, node types and filopodia ID. Those labels can affect nodes and edges or only nodes. The FE shows the components of the graph color-coded by the labels. There are seven mandatory and one optional label groups explained in the following subsections. If the graph has all seven mandatory label groups, it is a filopodia graph and only in this case all filopodia tools and functions are available.



Name	Color	Show	Location
⊞ TimeStep	⬛	☑	GRAPH
⊟ NodeType	⬛	☑	NODE
TIP_NODE	🟧	☑	
BRANCHING_NODE	🟨	□	
ROOT_NODE	🟦	□	
BASE_NODE	🟩	☑	
⊟ GrowthCones	⬛	☑	GRAPH
Object_001	🟧	□	
Object_002	🟦	☑	
Object_003	🟥	□	
⊟ ManualGeometry	⬛	☑	GRAPH
AUTOMATIC	🟥	☑	
MANUAL	🟩	☑	
⊞ ManualNodeMatch	⬛	☑	NODE
⊟ Location	⬛	☑	GRAPH
INSIDE	⬜	□	
OUTSIDE	🟦	☑	
⊟ Filopodia	⬛	☑	GRAPH
UNASSIGNED	⬜	□	
IGNORED	🟦	□	
AXON	⬜	□	
F_0000	🟥	□	
F_0001	🟩	□	
F_0002	🟩	☑	
F_0003	🟦	☑	
F_0004	🟪	☑	
F_0005	🟧	☑	
F_0006	🟥	☑	
F_0007	🟪	☑	
F_0008	🟧	☑	
F_0009	🟫	□	
F_0010	🟨	☑	
F_0011	🟦	□	
F_0012	🟧	□	
F_0013	🟦	□	
F_0014	🟥	□	
F_0015	🟪	□	
F_0016	🟦	□	
F_0017	🟥	□	
F_0018	🟥	☑	
F_0019	🟪	□	
F_0020	🟩	☑	

Figure 5: The labels of the filopodia graph are listed in the FE. They are necessary for view options, graph consistency and statistical analyses.

GrowthCones

This label group identifies which axon terminal is represented. It is important only during growth cone tracing and preprocessing. When starting the filopodia workflow the user identifies the axon terminals of interest by defining nodes in each axon terminal center. When starting propagation, those root nodes get their axon terminal ID with label name GC_xxx. All succeeding root nodes get the same ID. The number of axon terminal labels is equal to the number of axon terminals. The color of the different labels is random. Although only root nodes are assigned, the axon terminal labels could also be extended to nodes and edges.

TimeStep

The skeleton graphs of all time steps are combined in one dataset. All nodes and edges obtain a time label to mark the time step of their occurrence. Nodes or edges without a time label are not allowed.

NodeType

As mentioned before, there are 4 different node types in a filopodia graph: root nodes (blue), tip nodes (orange), base nodes (green) and branching nodes (yellow). This information is stored in the label group *NodeType*, which only handles nodes and not edges. Nodes without any node-type information are not allowed.

Location

For filopodia dynamics analysis, only the part between the tip and base node is significant since this is the actual filopodium. Everything else is the axon terminal body and is less important for our analyses. To differentiate between what is interesting and what is not, there is a label group called *Location*. It affects nodes and edges and has two possible labels: *GROWTHCONE* and *FILOPODIUM*. Edges and nodes that are part of the path between tip and base are labeled *FILOPODIUM*, those between base and root *GROWTHCONE*. The base is always labeled as *FILOPODIUM*. The filopodia toolbox offers to show only the components labeled as *FILOPODIUM* by enabling show *only filopodia* to make it clearer.

Filopodia

To distinguish between several filopodia, there is a label group called *Filopodia*. This group is essential for filopodia tracking over time. Nodes and edges can be assigned to this label group. During initialization of the filopodia graph, three default labels are defined: *Unassigned*, *Ignored* and *Axon*. In the beginning, all components (edges and nodes) are assigned to *Unassigned*. The user can assign components to *Axon* if they are part of the axon. This option is not used at the moment but might be interesting in future projects. Components assigned to *Ignored* will not be considered in further filopodia tracking. After triggering the filopodia tracking (see filopodia tracking) all components labeled as *GROWTHCONE* are assigned automatically to *Ignored* since they are not part of filopodia and irrelevant for the filopodia tracking. Depending on the tracking results, the base nodes and their connected filopodium part get their filopodia label with name base F_xxxx. The number of labels depends on the total number of filopodia. At the end of reconstruction and correction, the number should equal the number of filopodia + 3 (due to the unassigned, ignored and axon labels).

ManualNodeMatch

This label group resembles the filopodia label group. It has the same default labels: *Unassigned*, *Ignored* and *Axon*, but it only affects the nodes. The filopodia label group is based on the filopodia tracking results. This label group is based on filopodia propagation and manual input. In the beginning, all nodes are labeled as *Unassigned*. During filopodia propagation, the propagated succeeding node and the predecessor get the same ManualNodeMatch ID with name base M_xxxx. In some cases, the user has to manually add a filopodium. In such case, the software tries to find a predecessor based on filopodia tracking. If this fails and two filaments are mismatched the user can manually correct it using the tool match selected filopodia. This manual input affects this label group. The number of labels depends on the total number of filopodia. During progress, the number can be higher than the number of filopodia labels. At the end of reconstruction and correction, the number should equal the number of filopodia + 3 (due to the *Unassigned*, *Ignored* and *Axon labels*).

ManualGeometry

With this label group, the user input is tracked. It contains two labels: *Manual* and *Automatic*. Nodes added by the system are assigned to *Automatic*, nodes added by the user to *Manual*. The system adds nodes during filopodia propagation and the user adds nodes by correcting or adding filopodia (see filopodia tools). This label group allows the comparison of manual and automatic input.

Bulbous

There is an optional label group called *Bulbous* that can be added at the end of the filopodia reconstruction, before calculating the statistics. It allows the user to mark filopodia as *Bulbous* when they are quite stable and their tip appears spherical. Those filopodia are supposed to become a synapse and a stable connection to other neurons. This label group has two labels: *Nonbulbous* and *Bulbous*. When adding this optional hierarchical label, all graph components are automatically labeled as *Nonbulbous*. For more information about how to handle the bulbous label, see bulbous label.

Name	Location	Specific labels	Color	Name base	# Labels
GrowthCones	graph	/	random	GC_xxx	# GC
TimeStep	graph	/	red shaded	T_xxx	60
NodeType	nodes	TIP_NODE BRANCHING_NODE ROOT_NODE BASE_NODE	orange yellow blue green	/	4
Location	graph	GROWTHCONE FILOPODIUM	light blue dark blue	/	2
Filopodia	graph	UNASSIGNED IGNORED AXON	random	F_xxxx	# filopodia + 3
ManualNodeMatch	nodes	UNASSIGNED IGNORED AXON	random	M_xxxx	# filopodia + 3
ManualGeometry	graph	AUTOMATIC MANUAL	red green	/	2
Bulbous (optional)	graph	NONBULBOUS BULBOUS	light blue dark blue	/	2

Tracing with Dijkstra

The path between two nodes is traced using an intensity-weighted *Dijkstra shortest path algorithm*. A Dijkstra graph consisting of nodes and weighted edges is constructed. The set of nodes consists of all voxel centers contained in the axis-aligned box bound by the two nodes, extended in all directions by 10 voxels. The set of edges consists of all connections between neighboring voxels using 26-neighborhood. The edge weight w between two nodes v_1 and v_2 having intensity I_1 and $I_2 \epsilon [0, 255]$ is defined as:

$$w_{ij} = ||v_i - v_j||_2 + \frac{2c}{I'_i + I'_j}, I'_k = \min(I_k, I_{\max})$$

It consists of the distance between the nodes and an intensity term penalizing dark voxels. The user-adjustable intensity weight $c = 15.0$ balances these terms.

To avoid that long paths through bright voxels are preferred over slightly shorter paths with darker foreground voxels, foreground intensity is capped at $I_{\max} = 100$ (user-adjustable).

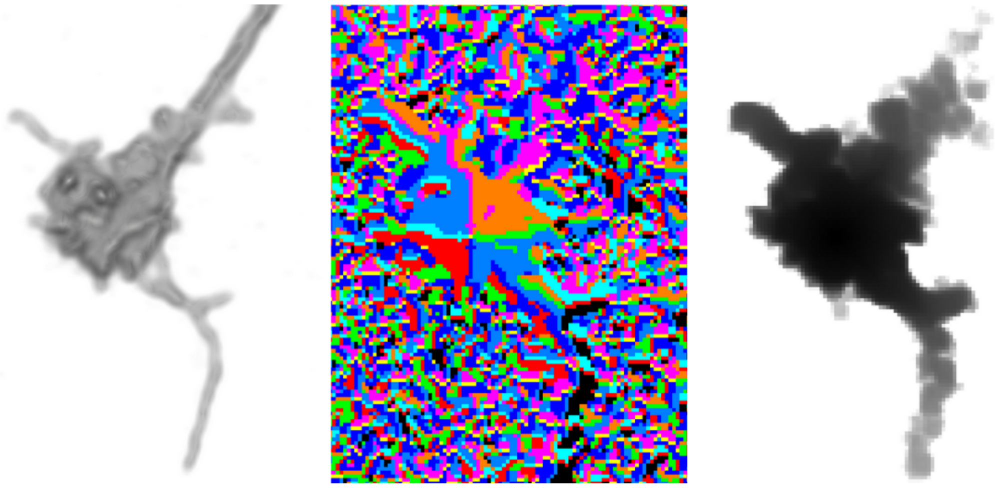


Figure 6: (a) Volume rendering of an axon terminal. (b) The prior map saves the neighboring prior voxel where the shortest calculated path came from using a 26-neighborhood. (c) Calculated distance from each voxel to the axon terminal center, whereby white means high distance and black small distance. Note that voxels related to the background have a high distance caused by the intensity term penalizing background voxels.

Tracing to root

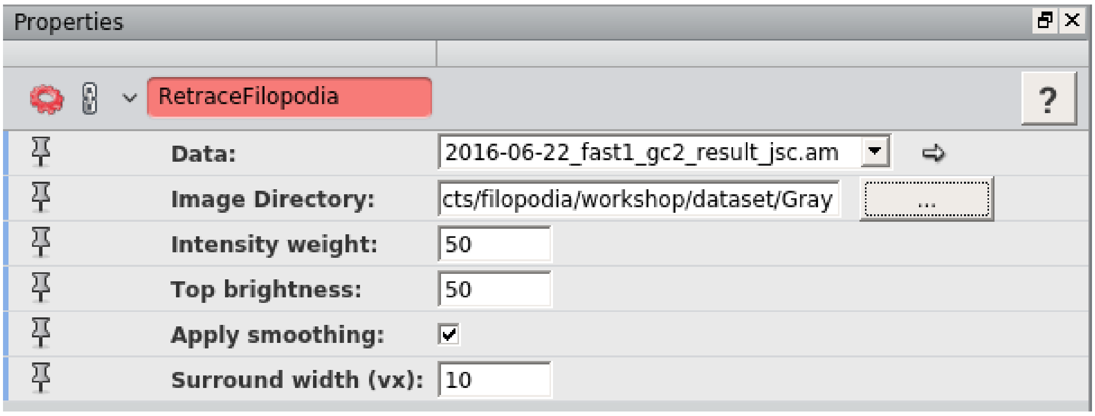
To speed up the common case of tracing from a point to the root node, Dijkstra maps storing the shortest path tree rooted at the axon terminal center are pre-computed. The tree is encoded by storing the 26-neighbor that is next on the shortest path to the root for each voxel. The shortest path from any voxel to the root can then be efficiently found by iteratively moving to the neighbor closest to the root until the root has been reached. To compute the path between any pair of non-root nodes, the Dijkstra graph is generated on-the-fly.

Intersections

To avoid coinciding paths, an intersection test with the existing edges of the graph is performed when adding a new path. For each newly traced edge point, proximity to an existing edge will be checked. Distance lower than the voxel size in z is considered as close. If there is a close edge point, it is taken as an intersection point. If the intersection point is not a source or target, it is converted into a new node, otherwise the intersection is an existing node. We assume that base nodes cannot be branching nodes since this is against the graph consistency. If the intersection is a base node the next edge point of the traced path is chosen. The latter is only possible if the edge has more than 2 edge points. In that case, the intersection is the adjacent node.

Retracing

Since the user-adjustable values intensity weight c and $I_{\max}$ need to be set during the data preparation part, it is hard to check whether appropriate values were chosen. For this reason, we provide the possibility to retrace all edges after the semi-automatic workflow using the Amira module *Retrace Filopodia* implemented in `hxfilopodia/HxRetraceFilopodia.cpp`. The user has to load the intensity images and define new values for c and $I_{\max}$. For each edge, the shortest path from source to target will be traced. Therefore, new Dijkstra graphs will be calculated on the fly with the new values. Additionally, the user can apply smoothing. The computation time depends on the number of edges.



Properties

RetraceFilopodia

Data: 2016-06-22_fast1_gc2_result_jsc.am

Image Directory: cts/filopodia/workshop/dataset/Gray

Intensity weight: 50

Top brightness: 50

Apply smoothing: ☑

Surround width (vx): 10

Figure 7: If traces are too smooth or jagged one can retrace the edges after finishing the filopodia tracking, using the module *RetraceFilopodia* tool. Image data, new *Intensity weight* and *Top brightness* have to be selected and set. All edges will be retraced, the nodes remain in the same positions.

Base node estimation

After tracing from a tip to the axon terminal center, the location of the filopodium base is automatically determined. The assumption is that the 2D intensity profile in a plane orthogonal to the traced path has a Gaussian shape for the filopodium part, and is non-Gaussian (ideally uniform) inside the axon terminal body. The base location is specified as the location where the intensity profile changes from Gaussian to non-Gaussian (Fig. 8 (c)). The Gaussian-ness is estimated at each point along the curve by computing the RMDS error between radially sampled intensity values in a 2D plane through the point orthogonal to the curve and a real Gaussian with parameters estimated from the sampled intensities. The RMSD error remains relatively constant inside the filopodium and increases sharply inside the axon terminal body. The base is positioned at the point where the RMSD increases.

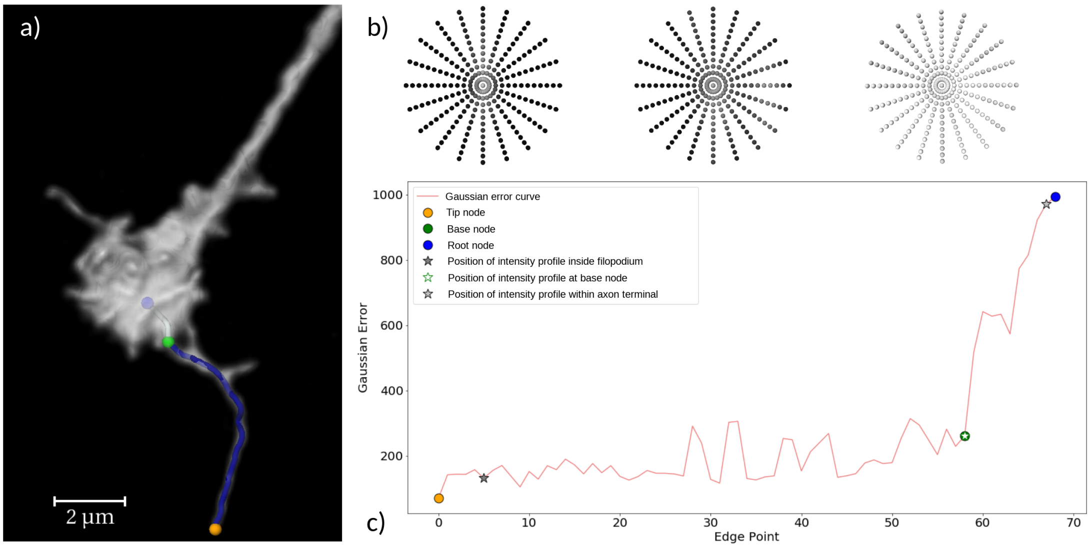


Figure 8: (a) The user adds a new filopodium by interactively specifying the tip (orange). The path to the root node (blue) is automatically traced. (b) The filopodia base is detected by computing the Gaussian-ness of the 2D intensity profile at each path point. From left to right: intensity profile from example point within filopodium, at base node (green), and from example point within axon terminal. (c) The change from Gaussian to non-Gaussian indicates the base. The peaks are caused by branches and the limited axial resolution in z.

Template matching algorithm

To propagate nodes during growth cone tracing or filopodia propagation, the template matching algorithm is used. The image region around the node in the previous time step is used as a template. The template center is shifted to all voxel centers within a search window in discrete steps. At each position, the normalized cross-correlation (NCC) is computed. The location with maximal NCC is the new node position. If the NCC is less than a chosen threshold $\gamma = 0.8 \in [0, 1]$, the new location is considered unreliable and the previous location is used. The template and search window size depend on the node type and are user-adjustable.

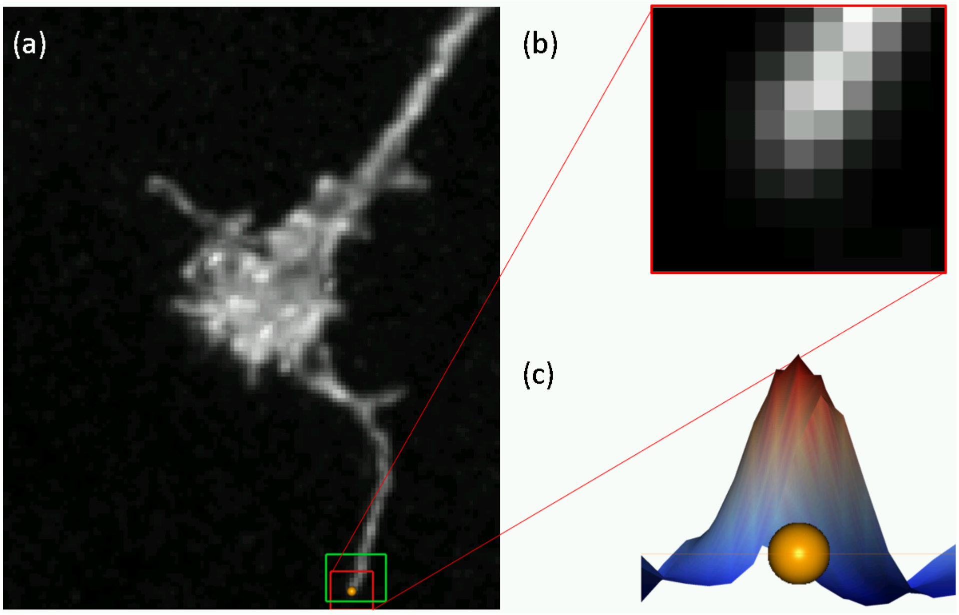


Figure 9: (a) To propagate the tip (orange) of a filopodium, the gray shape (red) around this tip is searched in the next time step by calculating the NCC at each voxel of a search area (green). Note that the center of the search window is corrected by the axon terminal center drift. (b) Gray shape (template) of a tip. (c) Height slice of the computed NCC within the search window. The position of the new node is the location of the maximal NCC.

Propagate root nodes

To propagate root nodes the template size $T_R \in \mathrm{IR}^3$ should be approximately the size of the axon terminal. The search window size $S_R \in \mathrm{IR}^3$ is set to the maximal expected movement of the axon terminal center. The following crop sizes should work for root nodes:

- $T_R = 3.5 \times 3.5 \times 1.5 \mu m^3$ $(35 \times 35 \times 3$ voxels)
- $S_R = 7.0 \times 7.0 \times 3.5 \mu m$ $(70 \times 70 \times 7$ voxels)

Propagate tip nodes

The template size $T_T \in \mathrm{IR}^3$ for tips is much smaller since the gray shape around a tip spans only a few voxels. But the search window $S_T \in \mathrm{IR}^3$ for the tips should be big enough to capture the agile filopodia ends. The following crop sizes should work for tips:

- $T_T = 0.5 \times 0.5 \times 1.0 \mu m^3$ $(5 \times 5 \times 4$ voxels)
- $S_T = 3.5 \times 3.5 \times 1.0 \mu m^3$ $(14 \times 14 \times 4$ voxels)

Propagate base nodes

To propagate base nodes, the template size $T_B \in \mathrm{IR}^3$ for bases is as big as the axon terminal body (axon terminal without filopodia). Since base nodes do not move much, the search window size $S_B \in \mathrm{IR}^3$ is small. The following crop sizes should work for tips:

- $T_B = 2.5 \times 2.5 \times 2.5 \mu m^3$ $(25 \times 25 \times 5$ voxels)
- $S_B = 1.0 \times 1.0 \times 1.0 \mu m^3$ $(10 \times 10 \times 2$ voxels)

Drifting and propagation

Axon terminals generally drift during image acquisition. Tips and bases are also influenced by this axon terminal drift. For this reason, the center of the search window is not the node position of the previous time step but the position corrected with the drift. r_{t-1} and r_t are the positions of the root nodes in time step $t-1$ and t.

Propagate filopodia

Filopodia propagation aims to detect successors of all filopodia of the current time step in the succeeding time step. For all filopodia, the tips and bases are propagated first. Second, paths between base and root, as well as between tips and base are traced using the algorithm described in section Tracing with Dijkstra. Finally, the newly found filopodium is assigned the same filopodia ID as the one in the previous time step.

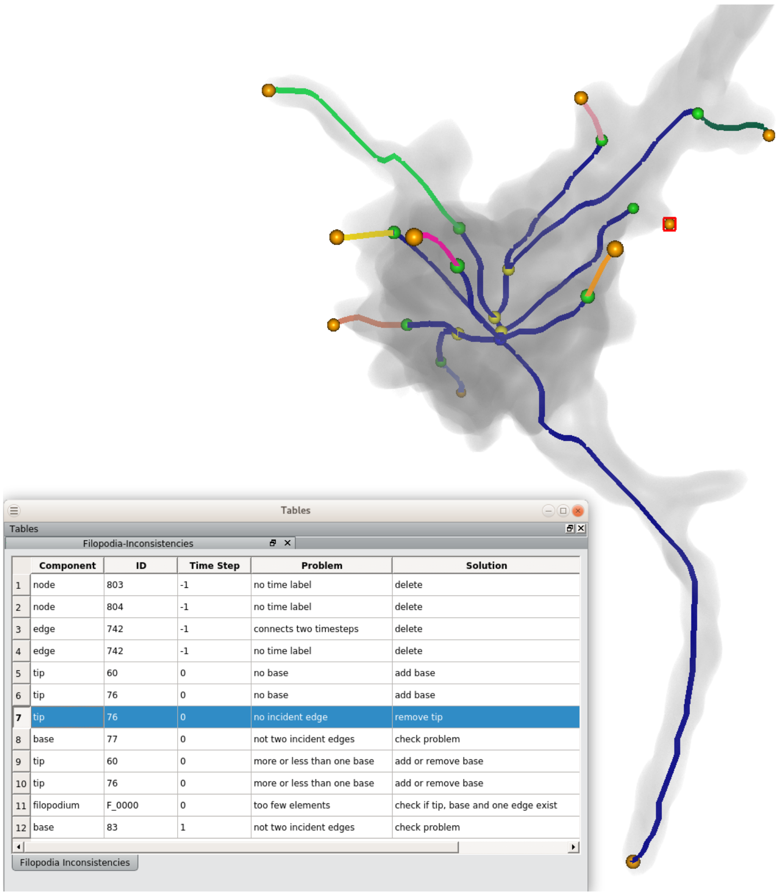
When the tip cannot be propagated (i.e. no location with NCC value above the threshold $\gamma = 0.8$), the filopodium is considered to have retracted. If no base can be found, the location of the base in the previous time step is used. If the length of the path between the base and tip is shorter than a user-specified threshold ($\lambda = 0.5 \mu m$), the filopodium is also considered retracted and is removed.

Graph Consistency

The filopodia graph should follow some constraints. Certain user behaviors can lead to inconsistencies. The user can check the consistencies by pressing the button *Check Consistency*. Performance depends on the number of nodes and edges in the graph since each component will be checked. Inconsistencies are listed in a table. The user can click on the rows of the table and the suspicious object will be highlighted in the filopodia graph. The viewer shows the suspicious time step if available. The table also gives a tip to get rid of the problem. Please resolve inconsistencies as often as possible, otherwise in some cases, the automatic parts won't work as expected. Propagation and updating statistics won't work unless all inconsistencies are fixed.

The following requirements have to be met:

- There is exactly one root node for each time step.
- Each tip has one base on the path to the root.
- Tip nodes have one incident edge, base nodes have two incident edges and branching nodes have 3 or more incident edges.
- Each filopodium has at least 2 nodes (one base and at least one tip), and at least one edge.
- Each match ID occurs once per time step.
- Filopodia exist continuously through the time steps (no gaps allowed).
- Nodes and edges have valid time labels.
- In each time step the axon terminal should be reconstructed as an independent graph without connections to other time steps.



	Component	ID	Time Step	Problem	Solution
1	node	803	-1	no time label	delete
2	node	804	-1	no time label	delete
3	edge	742	-1	connects two timesteps	delete
4	edge	742	-1	no time label	delete
5	tip	60	0	no base	add base
6	tip	76	0	no base	add base
7	tip	76	0	no incident edge	remove tip
8	base	77	0	not two incident edges	check problem
9	tip	60	0	more or less than one base	add or remove base
10	tip	76	0	more or less than one base	add or remove base
11	filopodium	F_0000	0	too few elements	check if tip, base and one edge exist
12	base	83	1	not two incident edges	check problem

Figure 10: After clicking *Check Consistency* the inconsistency table gives a list of inconsistent graph components based on the expected graph geometry. It shows the component type, the ID, the time step, the problem and how to solve it. By toggling a row, the suspicious graph element is selected and the viewer jumps to the inconsistent time step. In this example, the tip node 76 in time step 0 is selected. Since it has no incident edge it is recommended to delete this node.

Filopodia tracking

For filopodia tracking, the label filopodia is introduced to the skeleton graph. All graph elements of a filopodium have the same ID each time step the filopodium occurs. Usually, filopodia are tracked during the filopodia propagation. Thereby, the successors of tips and bases are detected in the following time steps. The new elements get the same filopodia ID as their priors. If the prior filopodium was unassigned, a new label ID is added. If the filopodia propagation fails, the user needs to add the successor manually. In such cases, the added filopodium gets its filopodia ID based on the point distances of the base nodes. This is possible because filopodia are identified by their base node since each filopodium has exactly one base per time step. Base nodes are not as agile as tips and mostly follow the general drift of the axon terminal.

Bulbous Label

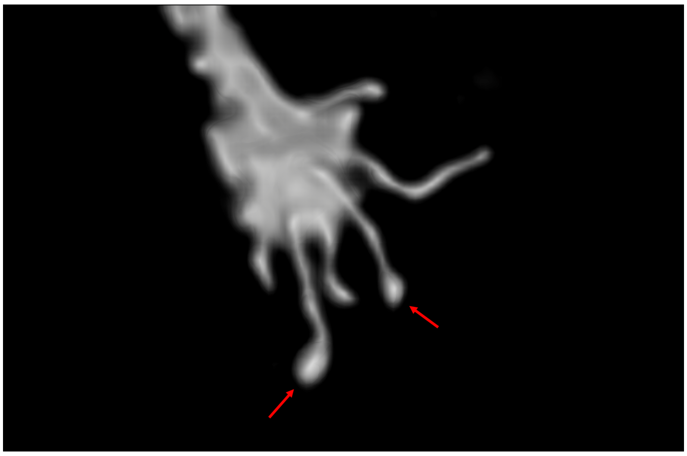


Figure 11: Example of bulbous filopodia. Those filopodia are usually more stable and the tip is nodular.

There is the possibility to add an optional label group called *Bulbous* (see Bulbous) by pressing *Add Bulbous Label Group* in the filopodia toolbox of the *Filament Editor*. A new label group with two labels *BULBOUS* and *NONBULBOUS* is added to the filopodia spatial graph. For all graph elements, the initial label is *NONBULBOUS*. A filopodium can be assigned to *BULBOUS* by selecting the base node, right-clicking on *BULBOUS* in the label tree and clicking *assign selection*. This can be repeated for all time steps this filopodium is bulbous. A more efficient way to assign the bulbous label to a filopodium in multiple time steps is:

1. assign base in the first time step the filopodium is bulbous to *BULBOUS*;
2. assign base in the last time step the filopodium is bulbous to *BULBOUS*;
3. press *Extend Bulbous Labels* button in filopodia toolbox of the FE.

By pressing *Extent Bulbous Labels*, all base nodes between the first and last bulbous time step are assigned too. It's not possible to have a bulbous filopodium for several time steps which becomes nonbulbous and later bulbous again. *Extend Bulbous Labels* won't work if the graph is inconsistent.

Since the bulbous label is not handled with a filopodia tool like *delete filopodia* or *trace filopodia*, it is not possible to undo the operation. Instead, the user has to assign the wrong parts to *NONBULBOUS* manually.

Statistics

After filopodia tracing and tracking, statistics can be generated by clicking *Update Statistics* in the filopodia toolbox of the FE. Three tables containing information of interests are generated. They are introduced in the following subsections.

FilamentStatistic

We call a filopodium in a single time step *Filament*. Filaments consist of one base node, at least one tip and a connecting edge. For each filament in the filopodia graph following information is stored in the table *FilamentStatistic*. *Unassigned* or *ignored* filaments are not stored.

Information	Explanation
Filament ID	ID for each filament

Information	Explanation
Time step	time step in which filament occur
Filopodia ID	Filopodia ID assigned to this filament
Filopodia name	Filopodia label name of Filopodia ID
Total length [μm]	Summarized edge length of all edges related to the filament
Angle	Angle of filopodia calculated by projecting root and base node of filopodium into XY plane and calculating the 2D angle between the vector spanned by root and base and the unit vector [0,1]
Bulbous	$b \in 0, 1$ indicates if filament is bulbous or non-bulbous
Base ID	Node ID of base
Base coords	x, y and z- coords of base
# Branches	Number of branching nodes
# Tips	Number of tips
Tip i ID	Node ID of tip for $i = 1, \ldots, n$; n is number of tips
Tip i Coords	x, y and z- coords of tip i;

FilopodiaStatistic

In *FilopodiaStatistic*, the information listed in the following table is stored for each filopodium. *UNASSIGMǸED* or *IGNORED* filopodia are not stored.

We call the change of length per time step *Speed*. For the length l of one filopodium in two time steps t and $t+1$ the speed is defined as $s = \frac{||l_{t+1} - l_t||}{\Delta t}$ with $\Delta t =$ 1 min.

Note that *Event* differs from *Lifetime*. Lifetime is the number of occurrences of the filopodium. An event is the change of filopodia length. If a filopodium occurs in the middle of the observation (not time step $t = 0$) there is an initial extension since the filopodia length changes from 0 to l. This is also true for the last occurrence: if a filopodium disappears during the observation (not in the last time step $t = 59$) there is a final retraction with filopodia length changing from l to 0.

Information	Explanation
Filopodia name	Filopodia label name
Filopodia ID	ID for each filopodium
# Bulbous	Total number of filaments with bulbous label
Bulbous [%]	Number of filaments with bulbous label related to lifetime

Information	**Explanation**
First occurence	Time step of first filament with filopodia ID
Last occurence	Time step of last filament with filopodia ID
Life time [min]	Number of filaments with filopdia ID
Length (mean) [μm]	Average length of filopodium
Length (std) [μm]	Std of filopodia length
Final length [μm]	Length of filopodium in last time step of occurence
# Total events	Total number of extension, retraction, and static events
Total speed (mean) [μm/min]	Average speed of filopodium
Total speed (std) [μm/min]	Std of filopodia speed
Filter = f	f is a user-adjustable threshold; Filopodia with speed less than f are static
# Filtered events	Number of events with speed higher than f
Filtered speed (mean) [μm/min]	Average speed higher than f
Filtered speed (std) [μm/min]	Std of speed higher than f
# Filtered extensions	Number of filtered events with $l_{t+1} - l_t$ > 0 (filopodium is longer than before)
Extensions speed (mean) [μm/min]	Average speed of extensions
Extentions speed (std) [μm/min]	Std of extension speed
# Filered rectraction	Number of filtered events with $l_{t+1} - l_t$ < 0 (filopodium is shorter than before)
Retraction speed (mean) [μm/min]	Average speed of retractions
Retraction speed (std) [μm/min]	Std of retraction speed
# Static	Number of events with speed less than f
Static [%]	Number of events with speed less than f related to toal number of events

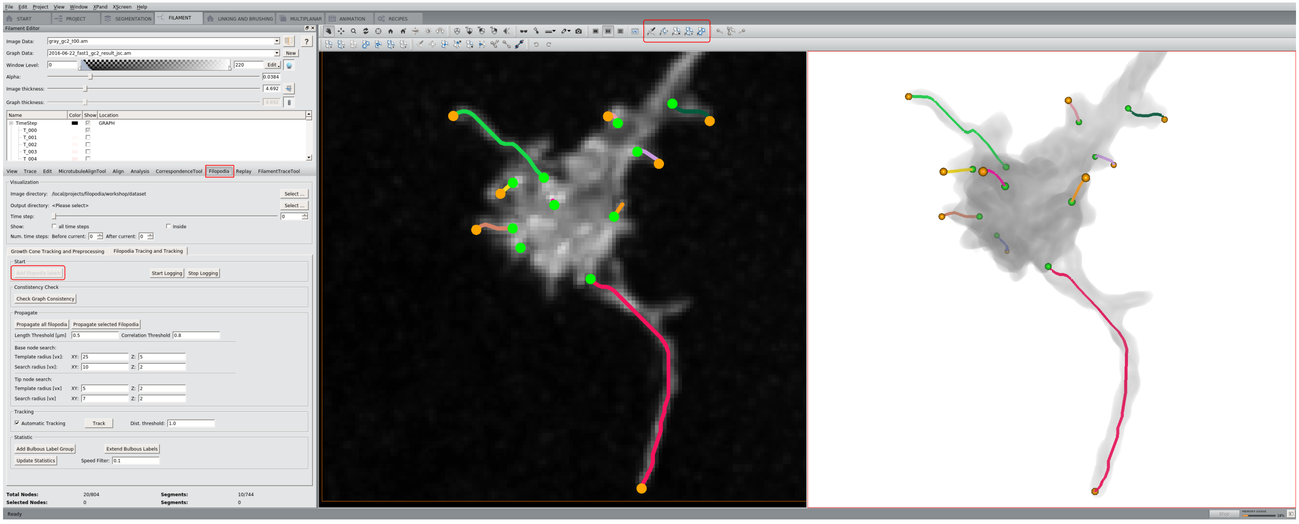
LengthStatistic

In this table, the filopodia length for all time steps is stored with the following additional information:

Information	Explanation
Filopodia name	Filopodia label name
Filopodia ID	ID for each filopodium
First occurence	Time step of first filament with filopodia ID
Last occurence	Time step of last filament with filopodia ID
Life time [min]	Number of filaments with filopdia ID
Angle	First angle of filopodia calculated by projecting root and base node of filopodium into XY plane and calculating the 2D angle between the vector spanned by root and base and the unit vector [0,1]
Filaments	List of filament IDs
Length t00 [μm]	Length of filopodium in time step t00
Length t01 [μm]	Length of filopodium in time step t01
...	...
Length t59 [μm]	Length of filopodium in time step t59

Filopodia Tools in Filament Editor

For the semi-automatic workflow for filopodia reconstruction, helpful tools are implemented in the *Filament Editor*. Those tools are marked with an *F* and are only available if the graph is a filopodia graph with the seven mandatory label groups (see Filopodia Labels).



Filament Editor

Image Data: gray_gc2_t00.am

Graph Data: 2016-06-22_fast1_gc2_result_jsc.am

Window Level: 0 – 220

Alpha: 0.0384

Image thickness: 4.692

Graph thickness: 4.692

Name	Color	Show	Location
TimeStep	■	☑	GRAPH
T_000		☑	
T_001		□	
T_002		□	
T_003		□	
T_004		□	

Visualization

Image directory: /local/projects/filopodia/workshop/dataset

Output directory: <Please select>

Time step: 0

Show: □ all time steps □ Inside

Num. time steps: Before current: 0 After current: 0

Propagate

Length Threshold [µm] 0.5 Correlation Threshold 0.8

Base node search:

Template radius [vx]: XY: 25 Z: 5

Search radius [vx]: XY: 10 Z: 2

Tip node search:

Template radius [vx] XY: 5 Z: 2

Search radius [vx] XY: 7 Z: 2

Tracking

☑ Automatic Tracking Dist. threshold: 1.0

Statistic

Speed Filter: 0.1

Total Nodes: 20/804 Segments: 10/744

Selected Nodes: 0 Segments: 0

Ready

Figure 12: The Filament Editor provides tools for the semi-automatic workflow, which are only available if the spatial graph has all filopodia labels. Press *Add Filopodia Labels* to add all missing label groups.

There are five filopodia tools listed in the table below: *Trace filopodia*, *Select filopodia*, *Match selected filopodia*, *Move filopodia base or tip* and *Delete filopodium*. The table gives an overview of the tools.

Tool name	Shortcut	Functions	Selection	Modifier
Trace filopodium	*shift* + *t*	define new filopodium move tip or base move base of a tip correct edge connect tip with edge	- tip or base tip edge tip	*shift* in 3D *shift* in 3D *shift* - *ctrl*
Select filopodium	-	select filopodia	node / edge node / edge	- *shift*
Match selected filopodia	-	isolate or match isolate or match with priors and successors isolate with successors	base(s) base(s) base	(*alt*) (*alt*) *shift* (*alt*) *ctrl*
Move filopodia base or tip	*shift* + *s*	add new base move tip or base along the edge	edge point tip or base + point	- -
Delete filopodium	*shift* + *d*	delete filopodium	tip	-

Trace filopodia

The *Trace Filopodia* tool is the most powerful tool of the filopodia workflow and offers many helpful options to create and correct a filopodia graph. It is implemented in `hxfilopodia/ExTraceFilopodia.cpp`. The tool requires grayscale images and Dijkstra maps to be loaded before. Make sure not to use the general tracing tool *Trace filament* since it won't provide all needed options nor keep the filopodia graph consistent. With a few minor exceptions, this tool is only available in the 2D view of the filament editor. Using the cursor in the 2D viewer changes the icon of the cursor. The following icons can occur:

- prohibit: screen coordinates can not be picked (invalid intensity or too many selected graph elements),
- seek: screen coordinates can be picked,
- seek with *N/E/P*: select node/edge/point,
- red seek icon: one graph element is already selected.

Depending on the selection, this tool allows adding new filopodia, moving nodes and correcting edges. Different options are explained in the following subsections.

Define new filopodium

Trace filopodia allows addition of new filopodia. The user primarily needs this function at the beginning of the filopodia reconstruction when all the filopodia have to be added manually, as well as in the following time steps, when one needs to add newly emerging filopodia.

To add new filopodia go to the 2D viewer of the Filament Editor. Search for a filopodium by scrolling through the image. Activate the *Trace Filopodia* tool (*shift* + *t*) and click on the position of the filopodia tip. Make sure that the cursor shows a white seek icon without any letter.

The current time step and the related root node will be estimated. Based on the precomputed Dijkstra map of that time step, the shortest path from the picked coordinates to the root node is estimated (read Tracing with Dijkstra for more information). During tracing the intersection with a graph element of the current time step is checked. The intersection is labeled as *GROWTHCONE* if there is no base node on the path from the intersection to the tip node, otherwise, it is labeled as *FILOPODIUM*. The following cases can occur and are handled respectively:

- Intersection is root node:
    - A new edge between tip and root
    - The intersection is labeled as *GROWTHCONE* (no base between tip and root yet)
- Intersection is node:
    - New edge between the tip and intersection node
    - Check if part of the axon terminal or filopodium
- Intersection is edge point:
    - Convert edge point to a node if convertible (not a node yet)
    - New edge between the tip and the converted point
    - Check if part of the axon terminal or filopodium

Please note that the intersection node is never a base. This would lead to a base node with three incident edges, which causes an inconsistent filopodia graph.

To provide a constant filopodia graph, all required labels are estimated. Afterward, the filopodia operation set *AddFilopodiumOperationSet* is called. It requires the new nodes and edges as well as their labels, and the intersection node or point. This operation set only calls the *MergeFilopodiaOperationSet*, which converts the connection point if necessary, adds all new nodes and edges, and assigns all the required labels.

The user can also define new filopodia in the 3D viewer. When picking the position of the filopodium tip one needs to hold the *shift* button. Unfortunately, the cursor does not offer the different icons and sometimes it seems like nothing happens after clicking. This happens when the intensity of the picked screen coordinate is invalid.

Move tip or base

The *Trace filopodia* tool can also be used to move a tip. This is useful if the user is unsatisfied with the position of added or propagated nodes.

First, the *Trace filopodia* tool (*shift* + *t*) has to be activated in the 2D viewer. Using the white crosshair one can select a base or a tip if the mouse pointer shows *N*. Afterwards, a new node position can be picked. The cursor should be a red crosshair since a node is already selected. When picking a screen position, while the crosshair shows an *N* or *E*, nothing will happen. If the seek icon shows a *P*, the edge point is selected and the node is moved to this edge point position. In such cases, the movement is operated similarly to *Move filopodia* base or tip* tool, described in Move tip or base along the edge. If the red crosshair is empty, the selected node is moved to the new screen coordinates. Depending on the node type, different steps are operated, described in the next sections.

Move tip

If the selected node is a tip, the position of the tip and the adjacent edge need to be updated. Therefore, the first adjacent edge and its source or target node (diverse from the tip node) are estimated. Then an edge from the new node position to the estimated neighboring node is computed using the Dijkstra algorithm. In this case, we do not provide intersections between the newly traced edge and the filopodia graph, but force the tip and the neighboring node to be the source and target of the new edge. Note that the precomputed Dijkstra maps can not be used since the adjacent node is the target and not the root node. This tool is also available in 3D viewer. Therefore, the user has to select the tip node that needs to be corrected and click on the new screen position while pressing *shift*. Unfortunately, the cursor does not offer the different icons and sometimes it seems like nothing happens after clicking. This happens when the intensity of the picked screen coordinate is invalid.

Move base

If the selected node is a base node, the position of the node, the path to the related tips and the path to the root need to be updated. First, all parts of the filopodia related to the selected base node are estimated. A temporary graph is initialized. All tips and the new base of the filopodia are added. The shortest path from the new base node position to the root node is computed, whereby intersections with filopodia graph components of that time step are allowed. The traced edge and the intersection are added to the temporary graph. Afterward, for each tip, the shortest path to the new base node position is traced using the Dijkstra algorithm. Intersections are only allowed to the components of the temporary graph. Note that the precomputed Dijkstra maps can not be used since the adjacent node is the target and not the root node. The temporary graph is then merged with the filopodia graph based on the estimated intersection node and deletes the old parts of the filopodium related to the selected base node as well as the old edge, which connects the base with the filopodia graph. If the new base position is too far away from the old base position, the software assumes a selection by mistake and stops the operation. This assumption is based on the fact that base nodes do not move much and for that reason, base node position corrections are not that distant.
This tool is not available in the 3D viewer.

Move base of a tip

To move a base, one can also select the related tip and click on a new base position by pressing *shift*. With this tool, the selected tip won't be moved, but the base of the selected tip will be. Some users prefer this option since it is easier to select tips than bases. Additionally, this can be used to split filopodia with two tips. A temporary graph is initialized and the selected tip and the new base are added. Afterward, the shortest path from the tip to the new base node position is traced using the Dijkstra algorithm. NNote that the precomputed Dijkstra maps can not be used since the adjacent node is the target and not the root node. The shortest path from the new base node position to the root node is computed, whereby intersections with filopodia graph components of that time step are allowed. The temporary graph is then merged with the filopodia graph based on the estimated intersection node.

Correct edge

With *Trace Filopodia* one can also correct the course of an edge by defining a supporting point. The selected edge will be traced through the selected point. This can prove helpful if an edge does not follow the filopodium well or if two filopodia intersect.

First, the user has to select the edge to correct in the 2D viewer. Activate the *Trace filopodia* tool (*shift* + *t*). One can also select the edge with this tool if the white crosshair shows an *E* inside. Afterwards, the user has to pick a supporting point by clicking on the screen coordinate. Note that it has to have a valid intensity, otherwise, the prohibition cursor is shown and the screen coordinate can not be selected. The source and the target node of the edge are estimated. Then, the path from the source to the supporting point and from the supporting point to the target is traced. Intersections are not provided. Finally, the new traced edge is added, the required labels are assigned, and the old edges are removed.

Connect tip with edge

Trace filopodia offers to connect a selected tip node with an edge point, which is not part of the shortest path from tip to root. This is helpful if filopodia are mistraced or to merge two filopodia into one.

In the 2D viewer, the tip to connect with an edge point of another path has to be selected. The *Trace filopodia* tool (*shift* + *t*) has to be active. The user can also select the tip with this tool if the white crosshair shows an *N* inside by scrolling over the node. Afterward, the edge point one wants to connect with has to be selected by pressing *ctrl*. The crosshair has to show *P* while selecting the edge point. The selected edge point is not allowed to be part of the shortest path from tip to root, since this wouldn't change anything. The selected edge point will later be converted into a branching node. A temporary graph is initialized. The selected tip as well as the converted edge point is added. The edge between the tip node and the new branching node is traced, whereby intersections are not provided. Afterward, it is checked if the selected edge point is labeled as *GROWTHCONE* (no base between edge point and root) or *FILOPODIUM* (base between edge point and root). If the label is *GROWTHCONE* a new base node has to be estimated (see Base node estimation) and added to the temporary graph. Afterward, all required labels are estimated.

Move filopodia base or tip

The tool *Move Filopodia Base or Tip* allows manual node corrections of bases or tips along an edge. Additionally, it provides the addition of a new base node on an edge. It is implemented in `hxfilopodia/QxMoveNodeTool.cpp`. This tool is helpful if the position of tips or bases are unsatisfying, but the traced edge is reasonable. It can also be used if a tip has no base node and needs to be added to make the filopodia graph consistent. It converts the selected edge point into a new node and removes the old one if necessary.

To select an edge point, the user needs to open the toolbox *View* and check *Points* to make them visible. We recommend disabling the point view as long as one does not need to select edge points for any tool.

Move tip or base along the edge

To move a base or a tip node, both, the node and an edge point, need to be selected. Activate the tool by pressing *Move Filopodia Base or Tip* or hitting *shift* + *s*. Afterwards, the following requirements will be checked:

- exactly one node and one edge point have to be selected,
- selected node has to be base or tip,
- node and edge point have to have the same *TimeStep* label ID,
- edge point needs to be convertible (not a node yet),
- if the tip node is selected, the edge point needs to be on the incident edge.

If the requirements are not fulfilled, an error message will occur.

When moving base nodes, the graph might become inconsistent. If the new base node is related to more than one tip, the graph becomes inconsistent since some tips might have more than one base. In such cases, the user has to convert one redundant base into an edge point. If the base was shared between two or more tips and the new position is now only related to one tip, the graph becomes inconsistent since the other tips have no base anymore. In such cases, the user has to add new base nodes afterward. This can also be realized with *Move Filopodia Base or Tip* tool.

Meanwhile, it is possible to use the *Trace Filopodia* tool to move nodes (see Move tip or base). Most of the users prefer it since it is more comfortable to use the same tool for many tasks. However, it is only available in 2D. Furthermore, the *Move Filopodia Base or Tip* tool provides more control and is more comprehensible since the user has to select the specific edge point and node and may have a better idea of where the node will move.

Add new base

To add a new base, the user needs to select the edge point on the position of the new base and trigger *Move filopodia base or Tip* or hit *shift* + *s*. The selected edge point is converted into a node and is assigned the required label.

Delete filopodium

The *Delete Filopodium* tool allows the removal of a filopodium in a single time step. This tool is implemented in `hxfilopodia/QxDeleteFilopodiaTool.cpp`. The *Delete Filopodium* tool is helpful if there is a redundant filopodium. This can happen if a filopodium was propagated although it has actually retracted and disappeared. It ensures that all parts of the filopodium are removed and provides a consistent filopodia graph. Standard *Delete selected nodes, edges, and points* tools should not be used since the graph may become inconsistent if the user does not select all parts of the filopodium.

To delete the extra filopodium the user should select the tip node of the filopodium and trigger the tool *Delete Filopodium* or hit *shift* + *d*. Before deletion, the fulfillment of the following requirements is checked:

- the graph is a filopodia graph, otherwise, the tool is not available,
- only one node is selected; no edge or edge point is selected,
- the selected node is a tip.

Not only the selected tip will be deleted, but also all parts of the filopodia graph to the closest branching node. First, the shortest path from the selected tip to the root node is estimated. If there is no base node on this path the graph is inconsistent and the operation stops. The path should contain at least one tip, one base and two edges. In this case, all components of the path but the root will be removed. If there are one or more branching nodes on the path, the closest branching node to the tip is estimated. The edges and nodes on the shortest path between the selected tip and the closest branching node are removed. The branching node will only be removed if it only has two incident edges left. If the filopodium has more than one tip, only the selected tip and the incident edge are removed. The branching node connecting at least one other tip is converted into an edge point if it has only two incident edges left.

Select filopodia

If the *Select Filopodia* tool is activated, the user can select all components of a filopodium at once. This tool is implemented in `hxfilopodia/QxSelectFilopodiaTool.cpp`. This tool is helpful if the user wants to select all components from base to tip of a filopodia in one or more time steps. This way it is possible to assign labels or similar for all selected graph elements at once.

To select a filopodium the user activates the tool *Select filopodium* and clicks on the node, edge, or edge point of the filopodium of interest. The *Filopodia* and *TimeStep* label of the selected graph element will be estimated and all elements with the same *Filopodia* and *TimeStep* label ID will be highlighted. While pressing *ctrl*, further filopodia can be added to the selection. With the modifier *shift*, the filopodia of all shown time steps will be selected. This means, that if all time steps are shown and the user selects a filopodium while pressing *shift*, all components with the same ID are highlighted.

Activating the tool *Select Filopodia* offers the following selections depending on the modifier:

- No modifier -> select filopodium in single time step,
- *ctrl* -> toggle filopodia in single time step,
- *shift* -> select filopodium in all time steps,
- *shift* + *ctrl* -> toggle filopodia in all time steps.

Match selected filopodia

The tool *Match Selected Filopodia* can be used to fix mismatches. This tool is implemented in `hxfilopodia/QxMatchFilopodiaTool.cpp`. It is helpful if a filopodium is added and is automatically assigned to the wrong filopodia ID.

To match or isolate filopodia, select the base nodes of interest. The user can also select segments or tips and the related base node is estimated automatically. The base nodes have to have successive *TimeStep* IDs without gaps. If a time step is missing in between, an error will occur and the task stops. If there are no gaps, the selected or related base nodes are given the identical *ManualNodeMatch* ID and they are assigned the same *Filopodia* ID during tracking.
In addition, with various modifiers priors and successors of selected nodes may also be assigned. The tool offers the following options:

- No modifier + n base nodes -> isolate or match only for selected time steps (selected nodes get identical new *ManualNodeMatch* ID),
- *shift* + 1 base node -> isolate with priors and successors (selected node and its priors and successors get new *ManualNodeMatch* ID),

- *ctrl* + 1 base node -> isolate with successors (selected node and its successors get new *ManualNodeMatch*ID (isolate with successors)),
- *shift* + 2 base nodes -> match with priors and successors (selected nodes, as well as the priors of the first node (smaller *timeStep* ID) and the successors of the second node (higher *TimeStep* ID), get new *ManualNodeMatch* ID),
- *ctrl* + >1 base node -> error,
- *shift* + >2 base nodes -> error.

By clicking the modifier *alt*, the user decides if the priors and successors are based on the *Filopodia* or *ManualNodeMatch* label. For the latter, one has to press *alt*. This option is a saver since only verified priors and successors are adapted but it might miss some obvious relatives. One shall not press *alt* to estimate priors and successors based on *Filopodia* label. This might adapt base nodes, which are not verified and might have the wrong *Filopodia* ID.