# Table of Contents

## 1 Associated paper

U-track 3D tackles on-going challenges in the interpretation and quantitative analysis of large arrangements of 3D trajectories as it arises with the measurement of intracellular dynamics with light-sheet micoscopy. The software is associated to the following publication currently under review.

Roudot, Philippe, Wesley R Legant, Qiongjing Zou, Kevin M Dean, Erik S Welf, Ana F David, Daniel W Gerlich, Reto Fiolka, Eric Betzig, and Gaudenz Danuser. "U-Track 3D: Measuring and Interrogating Intracellular Dynamics in Three Dimensions". BioRxiv. https://doi.org/10.1101/2020.11.30.404814.

## 2 Accessing the code and example dataset

Download the code:

```
git clone https://github.com/DanuserLab/u-track3D.git
```

Download the example datasets:

Endocytosis dataset: Breast cancer cells expressing eGFP-labelled alpha subunit of the AP-2 complex imaged with diaSLM by K. Dean (Dean et. al. 2016). The raw data has been cropped and limited to 50 time point (540MB).

Mitosis dataset: HeLa cells undergoing mitosis and expressing eGFP-labeled EB3 and mCherry-labeled CENPA imaged in dual-channel lattice light-sheet microscopy by W. Legant (David et. al. 2019). The raw data has been cropped, the entire sequence has been made available.

## 3 Script usage

Scripting is generally recommended for the analysis of a large number of acquisitions due to its flexibility and automated rendering capabilities. At the moment, the scripting library also provides more features than the GUI:

- A larger set of dynROIs are available
- Trackability computations
- Fully automated mipping overlay and video production
- Amira export of detection and tracking
- Point cloud rendering for faster display of detection mask

After cloning the repo and adding the code folder path to Matlab, you will find the following tutorial script in the repository:

```
tutoscript/trackingEndocyticPits_visualLibraryDemo.m
```

This script follows the sequence of processes possible in the GUI with additional visualization approaches:

- Detection and tracking on the full datasets
- Generate a dynROI following a set of trajectories
- Detection and tracking inside the dynROI
- Rendering results on MIP
- Exporting to Amira
- Point cloud rendering

**Example dataset**: Endocytosis dataset

```
tutoscript/trackingEndocyticPits_trackability.m
```

This script is a variation of the former to demonstrate manual dynROI selection and trackability:

- Detection and tracking on the full dataset
- Generate a dynROI from a set of trajectories
- Selecting manually a ROI within the DynROI
- Detection and tracking inside the manual dynROI
- Compute trackability in the manual dynROI
- Rendering results on MIP

**Example dataset**: Endocytosis dataset

```
tutoscript/trackingKFiberDynROI.m
```

This script demonstrates how to built a custom dynROI following two trajectories:

- Detection and tracking on the full dataset
- Generate a conical dynROI from a pair of trajectories
- Detection inside the custom dynROI
- Visualizaton and statistics

**Example dataset**: Mitosis datasets

The minimum configuration for each script includes:

- Adjusting parpool(#) with # the number of parallelizable core on your machine
- Adjusting <fullpath#> to the raw data folder name with # reflecting the number of channels
- Adjusting <analysisRoot> to your output folder name

# 4 Reproducing the u-track 3D paper with the tutorial scripts

The scripts reproduce the majority of features used in numerical experiments presented in the u-track 3D paper. Because of the large size of the datasets used in the paper, the script operates on smaller datasets that are amenable to direct download by the community. Here is the detail of the features that are demonstrated, their associated figures in the paper, and the script that reproduces them. One script can demonstrate several feature with a same dataset:

| Feature | Figure | Script |
|---|---|---|
| Tracking Brownian motions described by endocytic events | Fig1.a | trackingEndocyticPits_visualLibraryDemo |
| Multiscale Detections (Pole, KT, plus-ends) | Fig2.f-m | trackingKFiberDynROI |
| Building dynROI around tracks | Fig2.a | trackingEndocyticPits_visualLibraryDemo |
| Building conical dynROI between MT Poles and Chromosomes | Fig2.j | trackingKFiberDynROI |
| Building Manual ROI nested in a dynROI | Fig3.f | trackingEndocyticPits_trackability |
| Detection restricted to a dynROI | Fig2.k-m | trackingKFiberDynROI |
| Tracking restricted to a dynROI | Fig3.f | trackingEndocyticPits_visualLibraryDemo |
| Trackability | Fig3.f-g | trackingEndocyticPits_trackability |

Here is a list of the measurements that cannot be reproduced with those scripts and datasets due to the size of the associated datasets:

- Enocytic events lifetime analysis on on Breast Cancer Cell imaging (Fig1.b,c)
- Microtubule instability response to biochemical inhibition (Fig1.e,g).
- Adhesion shape studies (Fig2.a-d).
- Interpolar plane estimation from prometaphase to metaphase (Fig 2.f-i.)
- Trackability in the complete volume of the Breast cancer-cell, a crop of this dataset is however provided (Fig3.f-h)

Scripts to be released

- Tracking SOX2 single molecule with lattice light-sheet microscopy.
- Tracking microtubule dynamic instability

# 5 Performances

The software is CPU-optimized and has been tested on the following machines.

| Computer main features | OS | script | Time (s) |
|---|---|---|---|
| Intel Xeon E5-2680 28 cores @ 2.4 GHz - 528 GB Ram | Linux | trackingEndocyticPits_visualLibraryDemo (incl. rendering) | 142 |
| Intel Xeon E5-2680 16 cores @ 2.7 GHz - 32 GB Ram | Linux | trackingEndocyticPits_visualLibraryDemo (incl. rendering) | 206 |

# 6 GUI walk-through (Beta version)

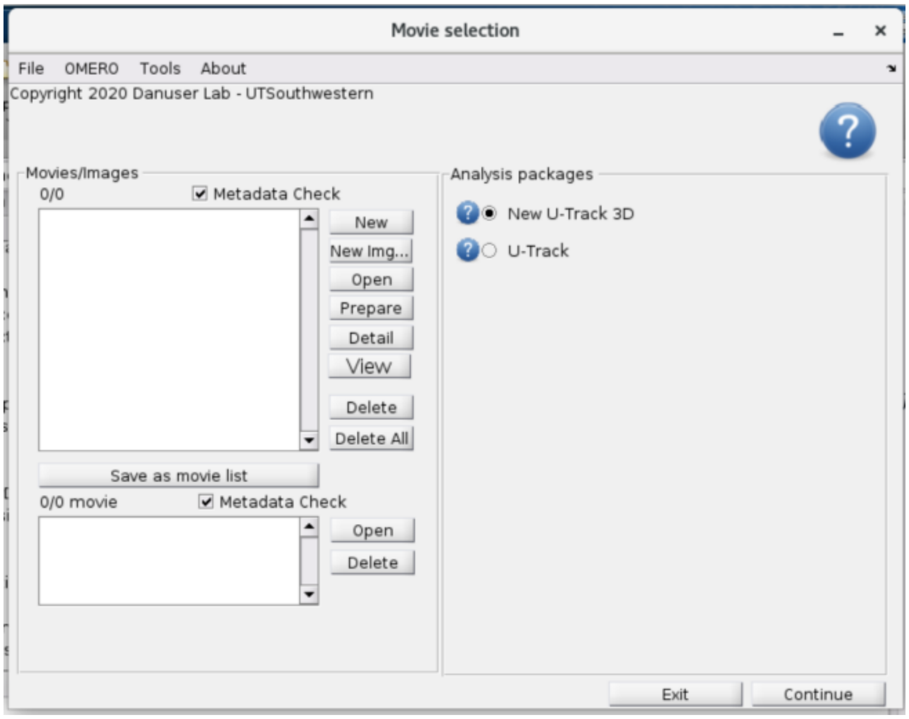
The GUI is generally recommended for the analysis of a couple files and test the capacity of u-track 3D on a given type of dataset. With straightforward data loading and a simplified execution pipeline, the GUI is designed toward an intuitive first experience.

## 6.1 Getting started and loading data

Create a parallel pool for parallel computing in matlab using either the command line or the graphical interface. Add the code folder in Matlab path. Then launch the GUI in the command line with:

```
movieSelectorGUI()
```

Then click "new" to create a new movie



Copyright 2020 Danuser Lab - UTSouthwestern

Movies/Images 0/0 ☑ Metadata Check

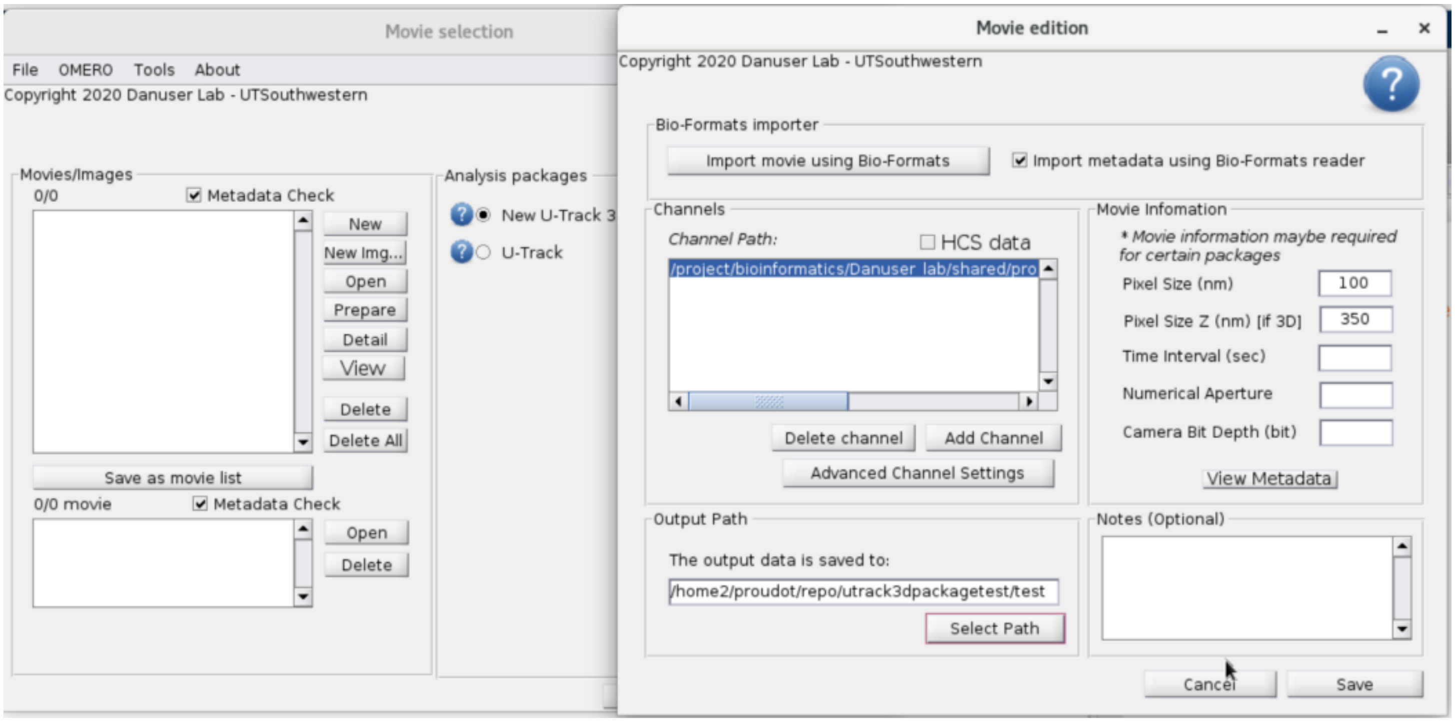
0/0 movie ☑ Metadata Check

Analysis packages

● New U-Track 3D

○ U-Track

If the data follows the bioformat standard, then open "Import Movie using Bioformat " and select the file, or first file of a sequence. If not, the format must be in a single tiff file per time point and each channel must be placed in a single folder. Use the "add channel" dialog to point to each channel folder



Copyright 2020 Danuser Lab - UTSouthwestern

Movies/Images 0/0 ☑ Metadata Check

0/0 movie ☑ Metadata Check

Analysis packages

● New U-Track 3

○ U-Track

Copyright 2020 Danuser Lab - UTSouthwestern

Bio-Formats importer

☑ Import metadata using Bio-Formats reader

Channels

Channel Path: □ HCS data

/project/bioinformatics/Danuser_lab/shared/pro

Movie Infomation

* Movie information maybe required for certain packages

Pixel Size (nm) 100

Pixel Size Z (nm) [if 3D] 350

Time Interval (sec)

Numerical Aperture

Camera Bit Depth (bit)

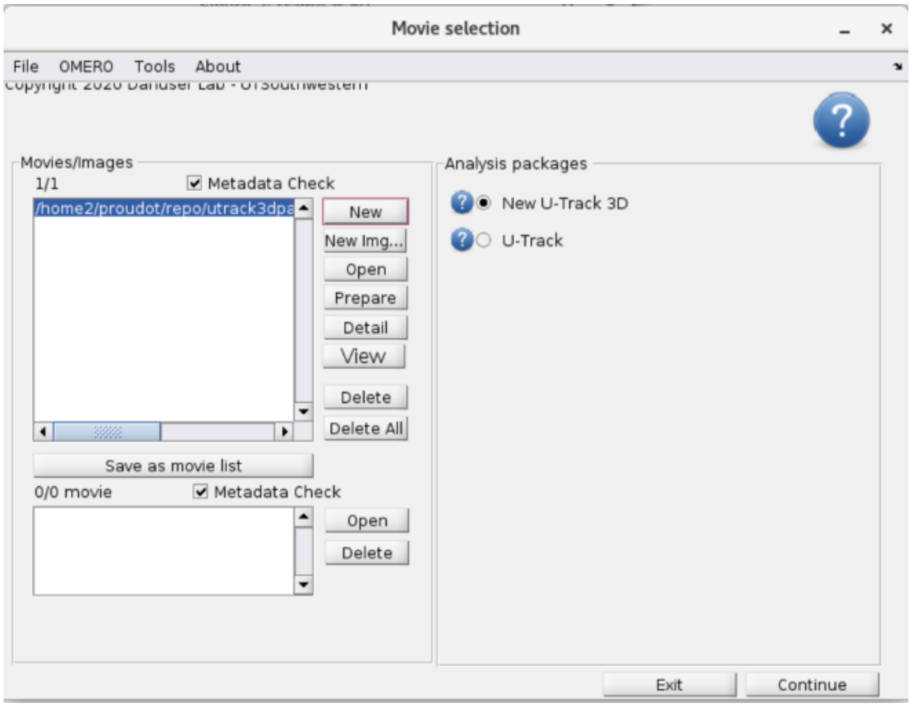
Output Path

The output data is saved to:

/home2/proudot/repo/utrack3dpackagetest/test

Notes (Optional)

Then launch the "New U-track 3D" application.



☑ Metadata Check

/home2/proudot/repo/utrack3dpa

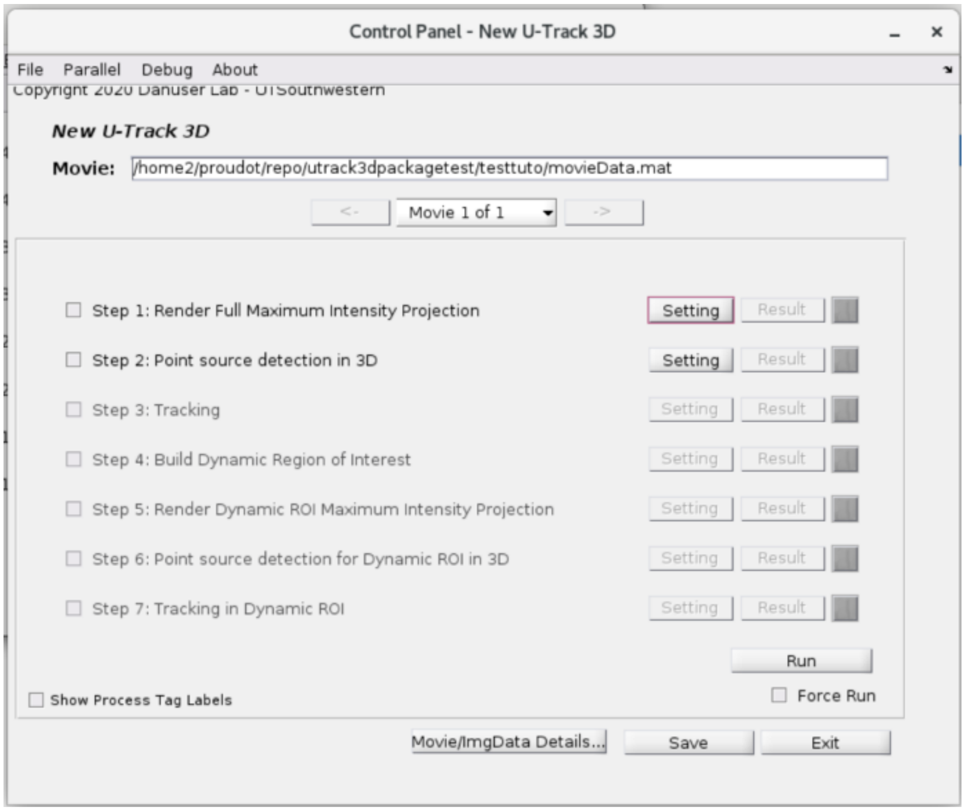
☑ Metadata Check

● New U-Track 3D

○ U-Track

In order to keep the set of operation linear, u-track 3D is organized in seven processes:

1. Maximum Intensity Projection (MIP) rendering
2. Detection on the full volume
3. Tracking on the full volume
4. Definition of a Dynamic Region Of Interest (DynROI)
5. Maximum Intensity Projection (MIP) rendering in the DynROI
6. Detection in the DynROI
7. Tracking in the DynROI



Movie: /home2/proudot/repo/utrack3dpackagetest/testtuto/movieData.mat

□ Step 1: Render Full Maximum Intensity Projection

□ Step 2: Point source detection in 3D

□ Step 3: Tracking

□ Step 4: Build Dynamic Region of Interest

□ Step 5: Render Dynamic ROI Maximum Intensity Projection

□ Step 6: Point source detection for Dynamic ROI in 3D

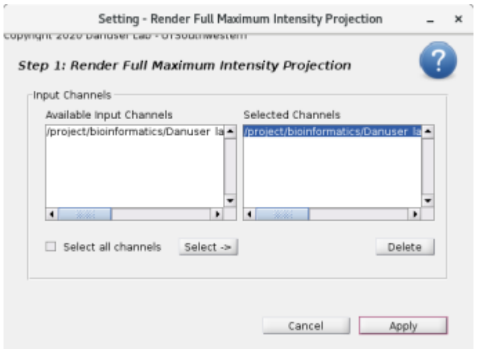
□ Step 7: Tracking in Dynamic ROI

□ Force Run

□ Show Process Tag Labels

## 6.2 Detection, tracking and review on the whole volumetric sequence

Each process must be parameterized or "setup" before being run. Sometime it merely involves opening the setup dialog and accepting the defaults by clicking "apply", as for example below with the MIP rendering process in the case of a single channel. This step ensures that the users explore the capacity the algorithm to adapt the parameters to their datasets.



Setting - Render Full Maximum Intensity Projection

Copyright 2020 Danuser Lab - UTSouthwestern

*Step 1: Render Full Maximum Intensity Projection*

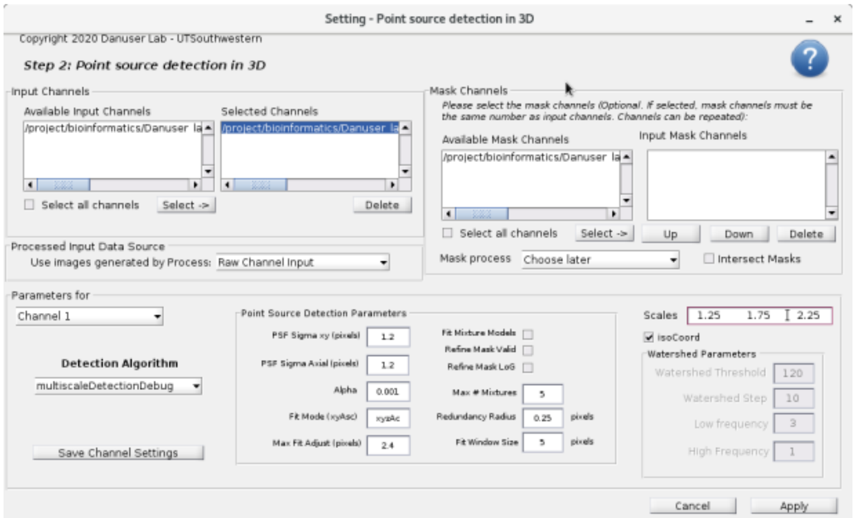
Input Channels

Available Input Channels: /project/bioinformatics/Danuser la

Selected Channels: /project/bioinformatics/Danuser la

□ Select all channels

The detection parameters propose different algorithms for detection, the default approach being the one presented in the u-track 3D paper. In this beta version, the "Scales" dialog define the scales used for filtering and the Alpha value define sensitivity. Further improvement will be made to separate different type of algorithms.



Setting - Point source detection in 3D

Copyright 2020 Danuser Lab - UTSouthwestern

*Step 2: Point source detection in 3D*

Input Channels

Available Input Channels: /project/bioinformatics/Danuser la

Selected Channels: /project/bioinformatics/Danuser la

□ Select all channels

Mask Channels

*Please select the mask channels (Optional. If selected, mask channels must be the same number as input channels. Channels can be repeated):*

Available Mask Channels: /project/bioinformatics/Danuser la

Input Mask Channels

□ Select all channels

Processed Input Data Source

Use images generated by Process: Raw Channel Input

Mask process: Choose later

□ Intersect Masks

Parameters for: Channel 1

**Detection Algorithm**: multiscaleDetectionDebug

Point Source Detection Parameters

PSF Sigma xy (pixels): 1.2

PSF Sigma Axial (pixels): 1.2

Alpha: 0.001

Fit Mode (xyAsc): xyzAc

Max Fit Adjust (pixels): 2.4

Fit Mixture Models □

Refine Mask Valid □

Refine Mask LoG □

Max # Mixtures: 5

Redundancy Radius: 0.25 pixels

Fit Window Size: 5 pixels

Scales: 1.25 1.75 2.25

☑ isoCoord

Watershed Parameters

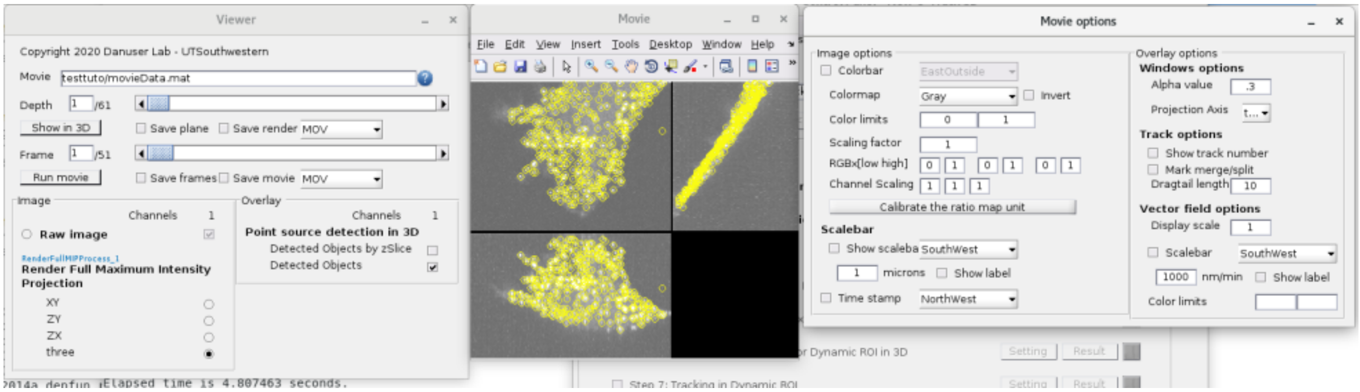
Watershed Threshold: 120

Watershed Step: 10

Low frequency: 3

High Frequency: 1

Click "Apply" and then "Run" in the control pannel to run the first two processes, then review the results by clicking on "Results" in step 2. Results can be seen overlayed over a MIP:



Viewer

Copyright 2020 Danuser Lab - UTSouthwestern

Movie: testtuto/movieData.mat

Depth: 1 /61

□ Save plane □ Save render MOV

Frame: 1 /51

□ Save frames □ Save movie MOV

Image

Channels 1

○ Raw image ☑

RenderFullMIPProcess_1

**Render Full Maximum Intensity Projection**

XY ○

ZY ○

ZX ○

three ●

Overlay

Channels 1

**Point source detection in 3D**

Detected Objects by zSlice □

Detected Objects ☑

```
2014a denfun :Elapsed time is 4.807463 seconds.
```

Movie options

Image options

□ Colorbar EastOutside

Colormap: Gray □ Invert

Color limits: 0 1

Scaling factor: 1

RGBx[low high]: 0 1 0 1 0 1

Channel Scaling: 1 1 1

**Scalebar**

□ Show scaleba SouthWest

1 microns □ Show label

□ Time stamp NorthWest

Overlay options

**Windows options**

Alpha value: .3

Projection Axis: t...

**Track options**

□ Show track number

□ Mark merge/split

Dragtail length: 10

**Vector field options**

Display scale: 1

□ Scalebar SouthWest

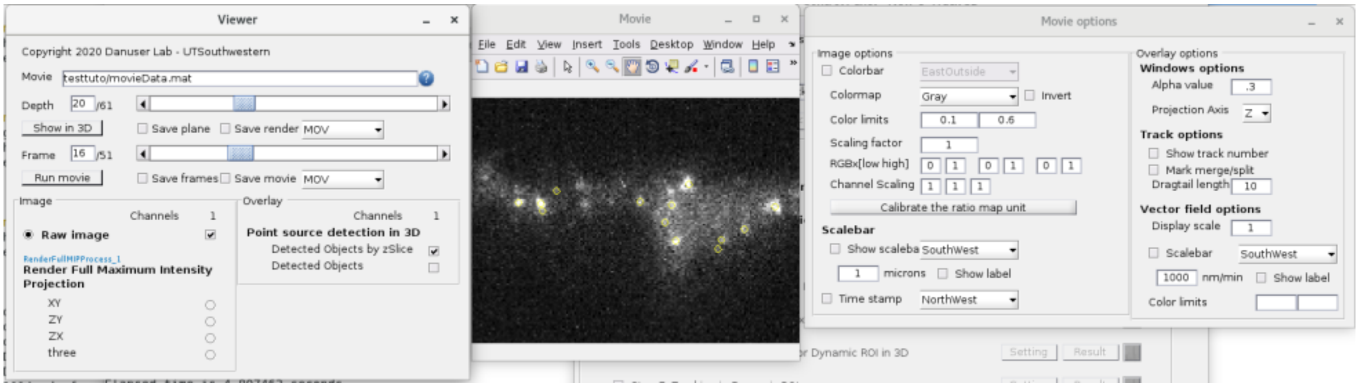
1000 nm/min □ Show label

Color limits

r Dynamic ROI in 3D

□ Step 7: Tracking in Dynamic ROI

Or by slicing the volume



Viewer

Copyright 2020 Danuser Lab - UTSouthwestern

Movie: testtuto/movieData.mat

Depth: 20 /61

Show in 3D | □ Save plane | □ Save render: MOV

Frame: 16 /51

Run movie | □ Save frames | □ Save movie: MOV

Image — Channels 1

● Raw image ☑

RenderFullMIPProcess_1

Render Full Maximum Intensity Projection

XY ○

ZY ○

ZX ○

three ○

Overlay — Channels 1

Point source detection in 3D

Detected Objects by zSlice ☑

Detected Objects □

Movie options

Image options

□ Colorbar: EastOutside

Colormap: Gray | □ Invert

Color limits: 0.1 | 0.6

Scaling factor: 1

RGBx[low high]: 0 1 | 0 1 | 0 1

Channel Scaling: 1 1 1

Calibrate the ratio map unit

Scalebar

□ Show scaleba: SouthWest

1 microns | □ Show label

□ Time stamp: NorthWest

Overlay options

Windows options

Alpha value: .3

Projection Axis: Z

Track options

□ Show track number

□ Mark merge/split

Dragtail length: 10

Vector field options

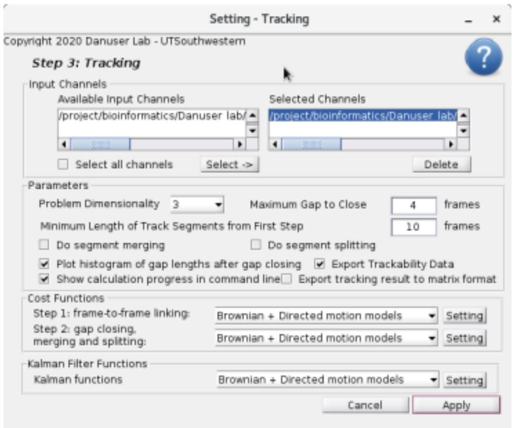
Display scale: 1

□ Scalebar: SouthWest

1000 nm/min | □ Show label

Color limits

The parameterization of the tracking algorithm first provides control over the maximum gap size, the minimum track size to be considered among other several controls.



Setting - Tracking

Copyright 2020 Danuser Lab - UTSouthwestern

Step 3: Tracking

Input Channels

Available Input Channels: /project/bioinformatics/Danuser lab/

Selected Channels: /project/bioinformatics/Danuser lab/

□ Select all channels | Select -> | Delete

Parameters

Problem Dimensionality: 3 | Maximum Gap to Close: 4 frames

Minimum Length of Track Segments from First Step: 10 frames

□ Do segment merging | □ Do segment splitting

☑ Plot histogram of gap lengths after gap closing | ☑ Export Trackability Data

☑ Show calculation progress in command line | □ Export tracking result to matrix format

Cost Functions

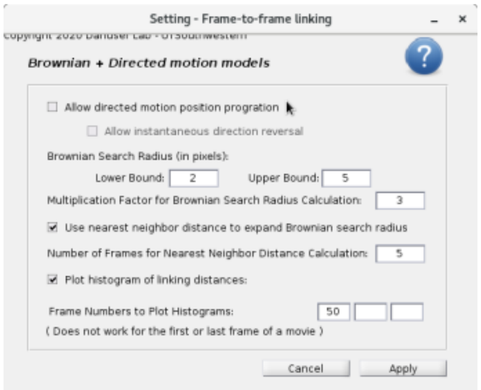
Step 1: frame-to-frame linking: Brownian + Directed motion models | Setting

Step 2: gap closing, merging and splitting: Brownian + Directed motion models | Setting

Kalman Filter Functions

Kalman functions: Brownian + Directed motion models | Setting

The control of Frame-to-frame linking and gap closing is performered in separated views:



Setting - Frame-to-frame linking

Copyright 2020 Danuser Lab - UTSouthwestern

Brownian + Directed motion models

□ Allow directed motion position progration

□ Allow instantaneous direction reversal

Brownian Search Radius (in pixels):

Lower Bound: 2 | Upper Bound: 5

Multiplication Factor for Brownian Search Radius Calculation: 3

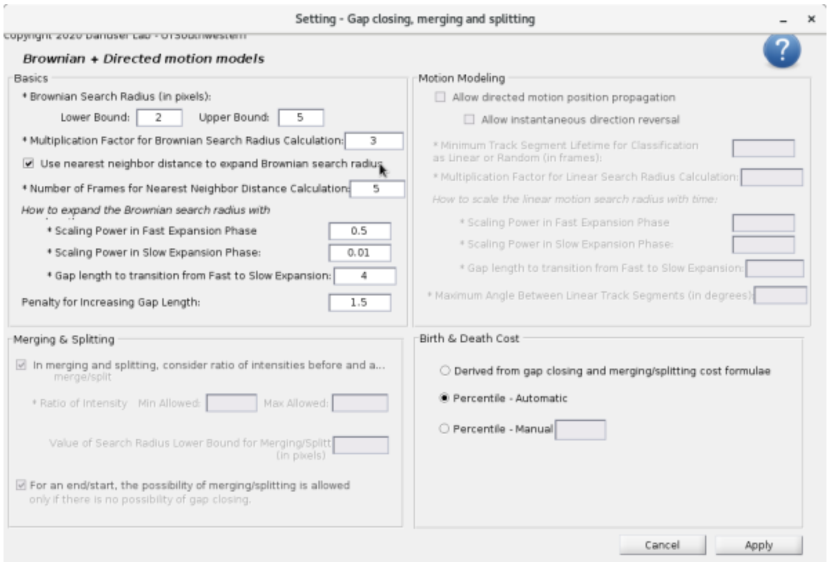
☑ Use nearest neighbor distance to expand Brownian search radius

Number of Frames for Nearest Neighbor Distance Calculation: 5

☑ Plot histogram of linking distances:

Frame Numbers to Plot Histograms: 50

( Does not work for the first or last frame of a movie )



Setting - Gap closing, merging and splitting

Copyright 2020 Danuser Lab - UTSouthwestern

*Brownian + Directed motion models*

Basics

- * Brownian Search Radius (in pixels): Lower Bound: 2 Upper Bound: 5
- * Multiplication Factor for Brownian Search Radius Calculation: 3
- ☑ Use nearest neighbor distance to expand Brownian search radius
- * Number of Frames for Nearest Neighbor Distance Calculation: 5

*How to expand the Brownian search radius with*

- * Scaling Power in Fast Expansion Phase: 0.5
- * Scaling Power in Slow Expansion Phase: 0.01
- * Gap length to transition from Fast to Slow Expansion: 4

Penalty for Increasing Gap Length: 1.5

Motion Modeling

- ☐ Allow directed motion position propagation
- ☐ Allow instantaneous direction reversal
- * Minimum Track Segment Lifetime for Classification as Linear or Random (in frames):
- * Multiplication Factor for Linear Search Radius Calculation:

*How to scale the linear motion search radius with time:*

- * Scaling Power in Fast Expansion Phase
- * Scaling Power in Slow Expansion Phase:
- * Gap length to transition from Fast to Slow Expansion:
- * Maximum Angle Between Linear Track Segments (in degrees):

Merging & Splitting

- ☑ In merging and splitting, consider ratio of intensities before and a... merge/split
- * Ratio of Intensity Min Allowed: Max Allowed:
- Value of Search Radius Lower Bound for Merging/Splitt (in pixels)
- ☑ For an end/start, the possibility of merging/splitting is allowed only if there is no possibility of gap closing.

Birth & Death Cost

- ○ Derived from gap closing and merging/splitting cost formulae
- ● Percentile - Automatic
- ○ Percentile - Manual

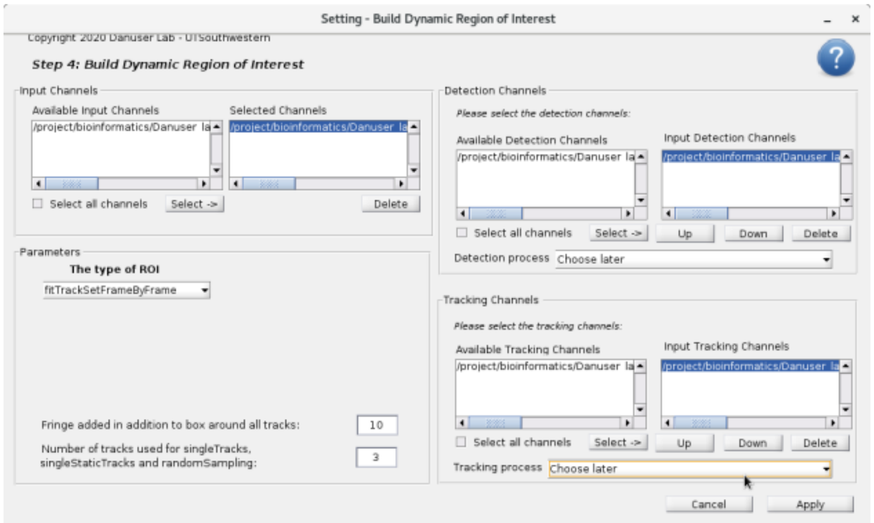
Once tracks are computed ("Apply" parameter and click on "run"), trajectories can be reviewed similarly to the review of detections.

### 6.3 Definition of DynROI and tracking in DynROI

The estimation of trajectories open the door to the dynROI in step 4. Several variation of dynROI estimation are made available such as:

- **fitTrackSetFrameByFrame**: fit an optimal box following a group of tracks over time. Motions are estimated on a frame-by-frame basis, ideal when the structural changes are important over time, but local changes are smooth over a few frames.
- **fitTrackSetRegistered**: fit an optimal box following a group of tracks over time. Motion are estimated with respect to the first frame, ideal when the structural changes are slow overtime but the local motions measured in the trajectories are highly stochastic.
- **fitDetSetFrameByFrame**: same as "fitTrackSetFrameByFrame" using detection instead of tracks.
- **fitDetSetRegister**: same as "fitTrackSetRegistered" using detection instead of tracks.

More dynROI types will be made available and documented in the near future.



Setting - Build Dynamic Region of Interest

Copyright 2020 Danuser Lab - UTSouthwestern

*Step 4: Build Dynamic Region of Interest*

Input Channels

- Available Input Channels: /project/bioinformatics/Danuser la
- Selected Channels: /project/bioinformatics/Danuser la
- ☐ Select all channels

Parameters

**The type of ROI**: fitTrackSetFrameByFrame

- Fringe added in addition to box around all tracks: 10
- Number of tracks used for singleTracks, singleStaticTracks and randomSampling: 3

Detection Channels

*Please select the detection channels:*

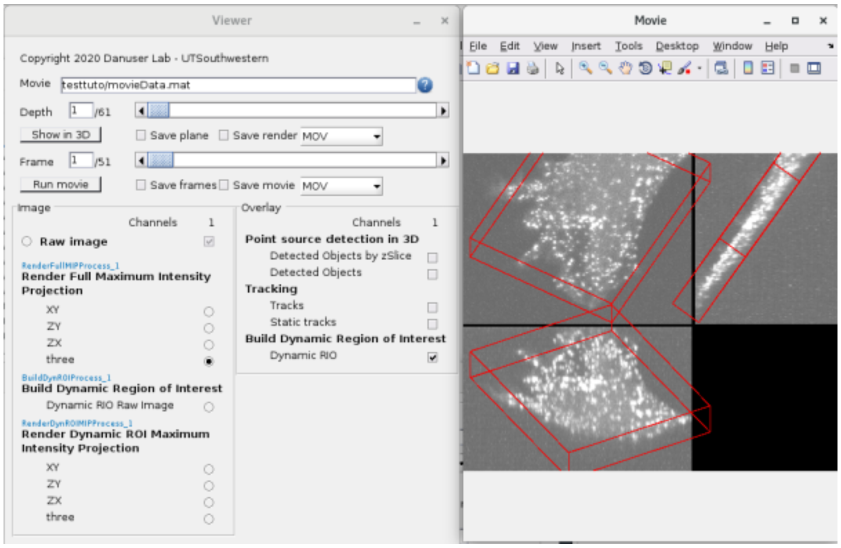
- Available Detection Channels: /project/bioinformatics/Danuser la
- Input Detection Channels: /project/bioinformatics/Danuser la
- ☐ Select all channels
- Detection process: Choose later

Tracking Channels

*Please select the tracking channels:*

- Available Tracking Channels: /project/bioinformatics/Danuser la
- Input Tracking Channels: /project/bioinformatics/Danuser la
- ☐ Select all channels
- Tracking process: Choose later

The review of dynROI location can be carried out using the MIP view in the Results panel:



Copyright 2020 Danuser Lab - UTSouthwestern

Movie testtuto/movieData.mat

Depth 1 /61

□ Save plane □ Save render MOV

Frame 1 /51

□ Save frames □ Save movie MOV

Image

Channels 1

○ Raw image ☑

RenderFullMIPProcess_1
Render Full Maximum Intensity Projection

XY ○
ZY ○
ZX ○
three ●

BuildDynROIProcess_1
Build Dynamic Region of Interest

Dynamic RIO Raw Image ○

RenderDynROIMIPProcess_1
Render Dynamic ROI Maximum Intensity Projection

XY ○
ZY ○
ZX ○
three ○

Overlay

Channels 1

Point source detection in 3D

Detected Objects by zSlice □
Detected Objects □

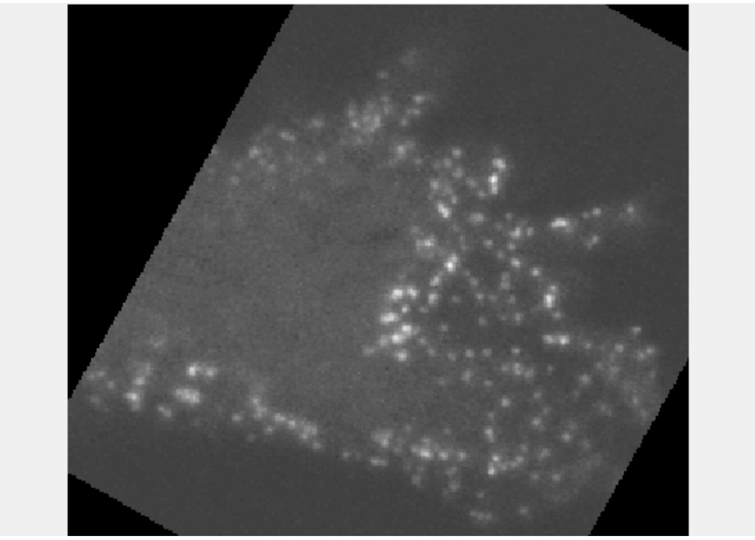
Tracking

Tracks □
Static tracks □

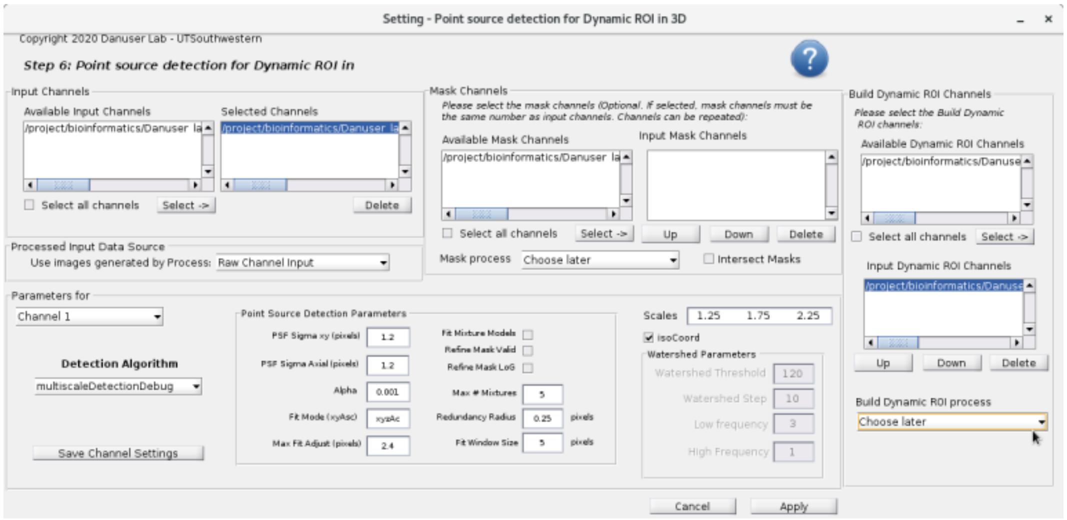
Build Dynamic Region of Interest

Dynamic RIO ☑

The voxels described by the dynROI can then be displayed by toggling on the "Dynamic ROI Raw Image" dialog. Here is a gif produced through the "save frames" dialog:



The last three remaining process 5 to 7 behave similarly to step 1 to 3, except that the dynROI built must be specified for the process to run properly using the "Build Dynamic ROI Process" drop-down menu as shown below for the detection process:


Setting - Point source detection for Dynamic ROI in 3D

Copyright 2020 Danuser Lab - UTSouthwestern

Step 6: Point source detection for Dynamic ROI in

Input Channels

Available Input Channels: /project/bioinformatics/Danuser la

Selected Channels: /project/bioinformatics/Danuser la

□ Select all channels

Mask Channels

*Please select the mask channels (Optional. If selected, mask channels must be the same number as input channels. Channels can be repeated):*

Available Mask Channels: /project/bioinformatics/Danuser la

Input Mask Channels:

□ Select all channels

Mask process: Choose later

□ Intersect Masks

Build Dynamic ROI Channels

*Please select the Build Dynamic ROI channels:*

Available Dynamic ROI Channels: /project/bioinformatics/Danuse

□ Select all channels

Input Dynamic ROI Channels: /project/bioinformatics/Danuse

Build Dynamic ROI process: Choose later

Processed Input Data Source

Use images generated by Process: Raw Channel Input

Parameters for: Channel 1

Detection Algorithm: multiscaleDetectionDebug

Point Source Detection Parameters

PSF Sigma xy (pixels): 1.2

PSF Sigma Axial (pixels): 1.2

Alpha: 0.001

Fit Mode (xyAsc): xyzAc

Max Fit Adjust (pixels): 2.4

Fit Mixture Models □

Refine Mask Valid □

Refine Mask LoG □

Max # Mixtures: 5

Redundancy Radius: 0.25 pixels

Fit Window Size: 5 pixels

Scales: 1.25 1.75 2.25

☑ isoCoord

Watershed Parameters

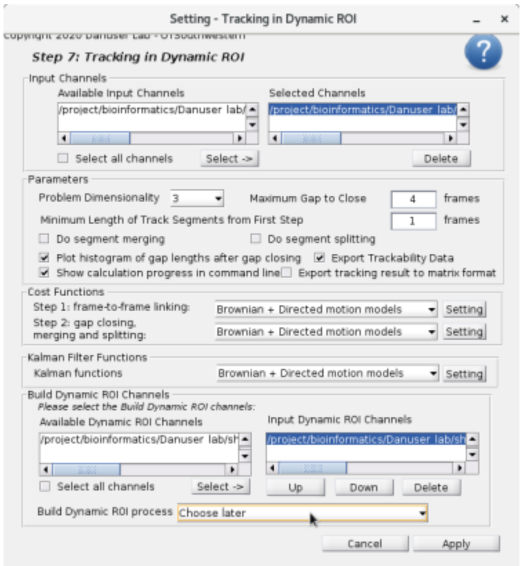
Watershed Threshold: 120

Watershed Step: 10

Low frequency: 3

High Frequency: 1

and for tracking process:


Setting - Tracking in Dynamic ROI

Copyright 2020 Danuser Lab - UTSouthwestern

Step 7: Tracking in Dynamic ROI

Input Channels

Available Input Channels: /project/bioinformatics/Danuser lab/

Selected Channels: /project/bioinformatics/Danuser lab/

□ Select all channels

Parameters

Problem Dimensionality: 3

Maximum Gap to Close: 4 frames

Minimum Length of Track Segments from First Step: 1 frames

□ Do segment merging

□ Do segment splitting

☑ Plot histogram of gap lengths after gap closing

☑ Export Trackability Data

☑ Show calculation progress in command line

□ Export tracking result to matrix format

Cost Functions

Step 1: frame-to-frame linking: Brownian + Directed motion models

Step 2: gap closing, merging and splitting: Brownian + Directed motion models

Kalman Filter Functions

Kalman functions: Brownian + Directed motion models

Build Dynamic ROI Channels

*Please select the Build Dynamic ROI channels:*

Available Dynamic ROI Channels: /project/bioinformatics/Danuser lab/sh

Input Dynamic ROI Channels: /project/bioinformatics/Danuser lab/sh

□ Select all channels

Build Dynamic ROI process: Choose later

## 7 Known issues

- The set of parameters for detection is confusing. Further streamlining improvement will be made for a smaller set of parameters to be visible.
- Detection and tracking sets used for dynROI estimation must be set manually, even when there is only one option.
- The Detection set must be selected before the last tracking step. This problem is also being solved.

## 8 Milestones

- `[X]` Adding Amira trajectory export in addition to detection in example script
- `[X]` Demonstrate trackability in the script
- `[ ]` Add an example of script-based Bioformt import
- `[ ]` Adding more types of DynROI in the GUI
- `[ ]` Fix annoying requirement in the GUI work flow (cf. Known Issues)

## 9 Software Requirements

- This software requires the following Matlab toolboxes
  - Matlab 2018a to 2019b
  - Computer Vision Toolbox
  - Image Processing Toolbox
  - Control System Toolbox
  - Optimization Toolbox

  - Image Processing Toolbox
  - Statistics and Machine Learning Toolbox
  - Curve Fitting Toolbox
  - Computer Vision Toolbox
  - Parallel Computing Toolbox

- This software has been on the following OS
  - Linux Red Hat 7

Created: 2020-12-15 Tue 01:56

Validate