Tracking Principles / Practicals

Kota Miura
Centre for Molecular and Cellular Imaging
EMBL Heidelberg
69120 Germany

miura@embl.de
http://cmci.embl.de

2014.07.08

Abstract

This lecture is an overview of tracking methods used in cell and developmental biology. Focus is made on the details of generally used two-step strategy by particle detection and particle linking. For those who are interested tutorials for using tracking plugins are also available in this note: manual tracking, kymograph and automated tracking.

1 Lecture Notes

Biological system is intrinsically dynamic. Time series of digital images, usually called a stack, contains temporal dynamics of position and intensity. By analyzing these dynamics we can extract numerical parameters which then enable us to characterize the biological system. It will then also be possible to characterize the event and trace back the timeline, to study the priori state. There are three types of dynamics.

1. Position does not change but intensity changes over time.
2. Position changes but the intensity does not change.
3. Both Position and Intensity change over time.

The lecture is about the measurement of type 2 dynamics generally called object tracking. Realistic situation demands measurement of type 3 *e.g.* you need to get a FRAP curve out of moving organelle, but we approximate the sample as type 2 as we need to learn the principle of tracking. Type 3 measurement will be your research that asks your ability to apply the full knowledge you acquired in the EMBL Advanced Imaging Course 2014, and probably some technique to use scripting languages to combine several tools.

For recent review on object tracking in cell and developmental biology, see Meijering et al. (2012). This article also contains an extensive list of available tracking packages. A bit old review on single particle tracking by Saxton is still an excellent textbook providing rich biophysical insights for analyzing tracking results (Saxton and Jacobson, 1997). For the classic "tracking" mentioned in the talk, or the presumptive fate map, see Vogt (1925) if you love biology (I am sure you do). This historical article could be downloaded from Google Books website. Magnificent movies produced by Keller *et al*. using single plane illumination microscopy and extensive image processing that almost killed the EMBL cluster by steaming it could be accessed through the web page of the paper in Science (Keller et al., 2008). Comparison of segmentation strategy in object tracking was published in 2001 by Cheezum et al. (2001) and has been a frequently cited paper in the field of biological object tracking. For overall review on image analysis methods see Hamilton (2009).

1.1 Dot Detection (Segmentation) techniques

- Manual tracking
  ImageJ "manual tracker plugin", more explanation in the tutorial section belwo.
- Manual contour tracing and centroid
- Thresholding
- Local intensity maxima
  - ParticleTracker plugin uses this method.
  - for more details, see the tutorial below.
- Gaussian-fitting
  - Vaccinia Virus tracking example in the talk.
  - Many applications for getting sub-pixel resolution coordinates.
  - You also should have heard about fitting for dot detection in super-resolution microscopy talks by Ricardo and Melike.
  - Aome available plugins for the Gaussian-fitting based tracking:
    * SpecleTrackerJ plugin
      · A tracker for fluorescence speckle microscopy.
      · `https://code.google.com/p/speckletrackerj/`
    * PTAj
      · A tracker for single molecule imaging. Uses minpak for the optimization.
      · `https://github.com/arayoshipta/projectPTAj`
    * The GDSC Single Molecule Light Microscopy (SMLM) plugins
      · For super-resolution microscopy (PALM/STORM) images, another one beside QuickPalm. It fits single molecule dots, recovers sub-pixel resolution coordinates of dots to render an super resolution image.
      · `http://www.sussex.ac.uk/gdsc/intranet/microscopy/imagej/smlm_plugins`
      · More tools for super-resolution microscopy are listed in `http://bigwww.epfl.ch/smlm/software/`
- Active Contour (SNAKES), Level-set
  - `http://iacl.ece.jhu.edu/projects/gvf/`
  - Quimp11, ImageJ Plugin
    `http://www2.warwick.ac.uk/fac/sci/systemsbiology/staff/bretschneider/quimp/`
- Machine Learning
  - There is no tracking package that does tracking using machine learning based segmentation, but it is a strong candidate in coming days.
  - Fiji plugin "Advanced Weka Segmentation"
    `http://fiji.sc/Advanced_Weka_Segmentation`
- Pattern matching (cross-correlation or sum-of-difference technique)
  - 3D tracking of macrophage-like cells.
    See Grabher et al. (2007)

1.2 Linking techniques

- Nearest-Neighbor Linking
  This is the simplest principle linking an object in one time point to the other.
- Evaluation of Overlaps
- Graph-theory based global optimization
  - Vallotton et al. (2003)
- Designing cost functions for the global optimization.
  - Sbalzarini and Koumoutsakos (2005)
  - Sbalzarini (2006)
- "Tracklets", a new way of dealing with dividing cells.
  - Jaqaman and Danuser (2009)
  - Bise et al. (2011)

1.3 Tracking Packages worth mentioning here.

For single molecule trackers, there is a pretty new plugin called PTAj, written by Yoshiyuki Arai. We do not try this plugin in the practical but for those involved in single molecule imaging, it might be worth trying it at your home institute. The plugin offers sophisticated interface to review the tracking results. We all know that even though the tracking is automatic, we start checking each track manually just to be sure, and just to be sure...

- `https://github.com/arayoshipta/projectPTAj`

There is also a new "all-in-one" tracking plugin for ImageJ. It's name is TrackMate, written by Jean-Yves Tinevez. The linking algorithm includes the most-up-to-date "Tracklets" and allows to track merging/splitting objects. I have had not time to test this plugin thoroughly, but the interface is similar to Imaris and should be easy for people who are used to wizard-style parameter setting for tracking. For developers, the attractive feature of this plugin is that it offers interface to implement your own algorithm for segmentation and linking.

- `http://fiji.sc/TrackMate`

2 Sample Sequences

Get Image Files

We use an example image sequence of EB1 labeled cells. EB1 is a microtubule associated protein that "moves" along microtubules. A file named **eb18_b.tif** is a sequence taken from single cultured cell labeled with eb1. This sequence is accessible by downloading an ImageJ plugin "Sample Image Loader". Instruction for installing this plugin is written in below.

Downloading A plugin for sample images

Download the plugin "Sample Image Loader" linked in the following page and install it.

- Download the plugin from `http://cmci.embl.de/downloads/sampleimageloader`
- Install the plugin using `[Plugin > Install...]`
- Restart Fiji, just to be sure with the installation.
- You should see a new menu item **EMBL**

The sequece can be opend by a menu command `[EMBL > Samples > eb1\_8b.tif]`. We try several different ways to measure movement of EB1 protein by tracking. Before tracking, examine the sequence first using stack related functions. Save the file as it is in your local machine, as it is not practical to download the image everytime when you want to open it.

Open the file in Fiji by

`[File > Open...]`

Check dimensions of the image by

`[Image > Properties...]`

...Image stacks are by default taken as a z-series and not t-series. Set Slices to 1, and Frames to appropriate size (number of frames).

Examine the sequence using stack tools.

Explore stack functions.
Start animation, Stop animation, change frame rates, Manipulations...These functions are located under

`[Image > Stack >]`

A useful plugin for stack

Running Z Projector plugin, written by Nico Stuurman, averages defined number of frames with moving window. This processing is powerful when you want to reduce noise from time-lapse sequences. Drawback is that some details of movement might be attenuated due to averaging. The plugin could be downloaded from

- `http://valelab.ucsf.edu/~nico/IJplugins/Running_ZProjector.html`

1. Download the plugin from above URL.
2. Install the plugin using `[Plugin > Install...]`
3. Restart Fiji, just to be sure with the installation.
4. Try running it on EB1 sequence.

3 Manual Tracking

In the simplest case, tracking can be done manually. The user can read out the coordinate position of the target object directly from the imaging software. In ImageJ, user can read out position coordinate indicated in the status bar by placing the cross-hair pointer over the object. Then coordinates can be listed in standard spreadsheet software such as Microsoft Excel or R for further analysis.

An ImageJ plug-in is also freely available to assist such simple way of tracking. An obvious disadvantage of the manual tracking is that the mouse-clicking by the user could be erroneous, as we are still human who gets tired after thousands of clicking. For such errors, measurement errors can be estimated by tracking the same object several times and this error could then be indicated together with the results. Otherwise, automated tracking is more objective, but if you have only limited number of tracks to analyze, manual tracking is a best choice before start to explore complex parameter space of automated tracking setting.

The manual tracking can be assisted by an ImageJ plug-in "Manual Tracking". This plug-in enables the user to record x-y coordinates of the position where the user clicked using mouse in each frame within a stack. The download site has a detailed instruction on how to use.

Download and Installation

Manual Tracking (Manual_Tracking.class) is a plugin bundled with Fiji. If you are using ImageJ, download the plugin file from the link below. The plugin was written by Fabrice Cordelieres.

- `http://rsbweb.nih.gov/ij/plugins/track/track.html`

MtrackJ is another manual tracker plugin that comes with Fiji. It has more functionalities and options than the *Manural Tracking* plugin (such as subpixel resolution estimates of clicked position when image is zoomed). We learn the manual tracking using the simpler plugin.

Measurements

1. Your task now is to track the movement of individual EB1 signal using the ManualTracker PlugIn. To ease detection of the signal, enhance contrast and convert the image stack from 16-bit to 8-bit.

   `[Image > Adjust > Brightness and Contrast]`

   and then convert to 8-bit

   `[Image > Type > 8bit]`

2. Then
   (a) Check "centering correction", use Local Maximum.
      - This allows to automatically snap the clicked position to the local maximum intensity.

   (b) Start manual tracking by clicking "Add track".
   (c) End tracking by "End Track"
   (d) Show tracks by "Drawing" Functions"

   You could track different particles by repeating steps between 2 and 3. Results window will list the measured positions for particles and step 4 will show a track overlaid image stack.
3. Tracked data can be saved by activating "Result" window and

   `[File > Save as...]`

4. **OPTIONAL:** Copy and paste the result table and plot the track in Excel or R and calculate a histogram of velocity distribution.

4 Kymograph

Kymographs are a two-dimensional time traces, where time *t* is in Y axis and space along a one-dimensional contour is in *X*, and the dynamical variable *F(x,t)* is visualized as an image. Kymographs provide a fast and convenient way to visualize motion and dynamics in microscopy images.We try measuring the speed of EB1 movement using Kymographs.

1. Open the image stack.
2. Contrast enhance,

   `[Image > Adjust > Brightness and Contrast]`

   and then convert to 8-bit

   `[Image > Type > 8bit]`

3. Do maximum intensity Z-projection

   `[Image > Stacks > Z-projection].`

4. Choose segmented line ROI tool (to do this, right click line-ROI icon in the menu bar. You will see a drop down selection). With projection image generated above, trace one of the tracks in the projection image.

5. Go back to the stack (click the window and activate it - or bring it to the front -),

   `[Edit > Selection > Restore Selection]`

   Then

   `[Image > Stacks > Reslice...]`

   Don't change parameters in the dialog window, simply OK. Resulting image is the kymograph.

6. To estimate the velocity of movement from kymograph, you simply need to measure the slope of the diagonal line shown in the kymograph. Since x-axis is distance and y-axis is time, if the starting point of the diagonal line is (x_1, y_1) and the end point of the line is (x_2, y_2) then the $slope = velocity$ is

   $$velocity = (x_2 - x_1)/(y_2 - y_1)$$

   Here is a simple macro to calculate this automatically from straight line selection.

   ImageJ macro for measuring kymograph `// scale and dt must be manually set`

```
var xyscale=0.207; //um
var dt=0.69; //sec

// sampling ROI coordinates, should be a straight line selection
print("-------------------");
getLine(x1, y1, x2, y2, lineWidth);
print("start ("+x1+" , "+y1+")   end ("+ x2 + " , " + y2 + ") ");

// calculation of speed.
dx=abs(x2-x1);
dy=abs(y2-y1);
dx *= xyscale;
dy *= dt;
velocity= dx/dy;
//output in log window.
print("dx = "+ dx +"um   during:" + dy + "sec");
print("Velocity = "+ d2s(velocity, 3) +" [um/s]");
```

   To use this macro, do

   `[File > New > Script]`
   to launch script editor, then select

   `[Language > ImageJ Macro]`
   in the script editor menu, copy and paste the code above. You could also copy and paste from the following page.

   - `http://cmci.embl.de/documents/121005advancedimg`

7. **OPTIONAL:** Collect your data in Excel or R and plot tracks and a histogram of velocity distribution.

8. **OPTIONAL:** For those interested in using kymograph intensively, there is a good plugin that assists you in doing so efficiently. The plugin was written by Jens Rietdorf and Arne Seitz

   - Multiple Kymograph Plugin for ImageJ
     `http://www.embl.de/eamnet/html/kymograph.html`

9. **OPTIONAL:** To quantify ambiguous kymograph (meaning hard to draw a line, texture like streakes) use:

   - Kymoquant macro for ImageJ
     `http://cmci.embl.de/downloads/kymoquant`

   Written by myself, with valuable suggestions from Peter Lenert.

5 Automated Tracking: ParticleTracker Plugin

This plugin uses linking algorithm that involves global optimization based on cost function that involves signal intensity and its second order derivative. The plugin was first written as a Matlab code by Ivo Sbalzarini and the migrated to ImageJ plugin by his group members, Guy Levy and Janick Cardinale.

Download ParticleTracker Plugin and Install

The plugin could be down loaded from

- `http://www.mosaic.ethz.ch/Downloads/ParticleTracker`

To install the downloaded plugin, `[Plugins > Install...]`. You could also directly copy the file to plugins folder under Fiji directory and do `[Help > Refresh Menus]`.

Just in case if the above plugin somehow does not work, try download and use the version linked in the flowing site.

- `http://cmci.embl.de/downloads/particletracker2d`

Start the ParticleTracker plugin

Start the particleTracker by `[Plugins > Mosaic > ParticleTracker 2D/3D]`.

Study Dot Detection Parameter

This tracking tool has two parts. First, all dots in each frame are detected, and then dots in successive frames are linked. The first task then is to determine three parameters for dot detection. There are three parameters.

1. **Radius**
   Expected diameter of dot to be detected in pixels.
2. **CutOff**
   Cutoff level for the none-particle discrimination criteria, a value for each dot that is based on intensity moment order 0 and 2.
3. **Percentile**
   Larger the value, more particles with dark intensity will be detected. Corresponds to the area proportion below intensity histogram in the upper part of the histogram.

Try setting different numbers for these parameters and click ”Preview Detected”. Red circles appear in the image stack. You could change the frames using the slider below the button.

After some trials, set parameters to what you think is optimum.

Set Linking parameters

Two parameters for linking detected dots should be set.

1. **Link Range**
   . . . could be more than 1, if you want to link dots that disappears and reappears. If not, set the value to 1.

2. **Displacement**
   ...expected maximum distance that dots could move from one frame to the next. Unit is in pixels.

After parameters are set, click "OK". Tracking starts.

Inspect the Tracking Results

When tracking is done, a new window titled "Results" appears. At the bottom of the window, there are many buttons. Click "Visualize all trajectories", and then a duplicate of the image stack overlaid with trajectories will appear.

Select a region within the stack using rectangular ROI tool and then click "Focus on Area". This will create another image stack, with only that region. Since this image is zoomed, you could carefully check if the tracking was successful or not.

If you think the tracking was not successful, then you should reset all the parameters and do the tracking again.

Export the tracking results

To analyze the results in R, data should be saved as a file. To do so, first click "All Trajectories to Table". Results table will then created. In this results table window, select `[File > Save As...]` and save the file on your desktop. By default, file type extension is ".xls", excel format, but change this to ".csv". CSV stands for "comma separated file", and this is more classic but general data format which you could easily import in many software including R.

Further analysis

OPTIONAL: Import your results from Excel or R and plot tracks and a histogram of velocity distribution.

6 Particle Image Velocimetry (PIV)

Download and Installation

There is a **PIV** plugin bundled with Fiji, but we use another PIV plugin developed by Qingzong Tseng. You need several files for this plugin to run. Here is the recipe for running the plugin in 64-bit Windows OS.

1. Go to
   - `https://sites.google.com/site/qingzongtseng/piv`

   and scroll the page down to "Installation" section. There you will find links to **PIV_.jar**, **javacv.jar** and **jna.jar** file. Download these three files.
2. Move or copy above three files to the **plugin folder** within Fiji or ImageJ folder.
3. Go to
   - `https://sites.google.com/site/qingzongtseng/template-matching-ij-plugin`

   (or you could click "template matching and slice alignment" link in the side bar in Zong's page) and scroll down to "Installation" section. There you will find a link to OpenCV library for 64-bit windows. Download the file and unzip it. You will find two files, cv100.dll and cxcore100.dll, then.
4. Move or copy above two files, cv100.dll and cxcore100.dll, to the Fiji or ImageJ folder (**not the plugin folder**).

Analysis

1. PIV plugin works only with two-frame stack. Reduce the stack size by Duplicating only part of the stack. To do so:

   - `[Image > Duplicate...]` then in the dialog panel, set the duplication range only two frames e.g. 10-11.

2. There are three different functions for doing PIV under `[Plugins > PIV >]`, Cross-Correlation, Basic and Advanced. Zong recommends not to use Cross-Correlation. We try to do PIV using "Basic".
3. When you choose `[Plugins > PIV > Basic]`, you will see a panel asking you to input parameters. There are three iterations, first start to estimate displacement of signal with large grids and then to finer grids. By starting from larger grids, estimation procedure recognizes global movement first and then more detailed movements in later iterations. Try to set windows size so that it fits the size of the image.
4. Final output is a text file. This could be saved anywhere, but be sure that you know where it is.
5. To plot the PIV field (Vectors), `[Plugins > PIV > Plot]`. Try to plot without changing the default value. If the plot looks bad with too large / small arrows, close the plot and do the plotting again with sums adjusted parameters.
6. **OPTIONAL** Since output is a text file of vector values, you could import the text file from Excel or R to plot histogram of velocities.

References

Ryoma Bise, Zhaozheng Yin, and Takeo Kanade. Reliable cell tracking by global data association. In *2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 1004–1010. IEEE, March 2011. ISBN 978-1-4244-4127-3. doi: 10.1109/ISBI.2011.5872571. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5872571http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true\&arnumber=5872571`.

M K Cheezum, W F Walker, and W H Guilford. Quantitative comparison of algorithms for tracking single fluorescent particles. *Biophysical journal*, 81(4):2378–88, October 2001. ISSN 0006-3495. doi: 10.1016/S0006-3495(01)75884-5. URL `http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1301708\&tool=pmcentrez\&rendertype=abstract`.

Clemens Grabher, Adam Cliffe, Kota Miura, Joel Hayflick, Rainer Pepperkok, Pernille Rø rth, and Joachim Wittbrodt. Birth and life of tissue macrophages and their migration in embryogenesis and inflammation in medaka. *Journal of leukocyte biology*, 81(1):263–71, January 2007. ISSN 0741-5400. doi: 10.1189/jlb.0806526. URL `http://www.ncbi.nlm.nih.gov/pubmed/17046968`.

Nicholas Hamilton. Quantification and its applications in fluorescent microscopy imaging. *Traffic (Copenhagen, Denmark)*, 10(8):951–61, August 2009. ISSN 1600-0854. doi: 10.1111/j.1600-0854.2009.00938.x. URL `http://www.ncbi.nlm.nih.gov/pubmed/19500318`.

Khuloud Jaqaman and Gaudenz Danuser. Computational image analysis of cellular dynamics: a case study based on particle tracking. *Cold Spring Harbor protocols*, 2009(12):pdb.top65, December 2009. ISSN 1559-6095. doi: 10.1101/pdb.top65. URL `http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3155779\&tool=pmcentrez\&rendertype=abstracthttp://cshprotocols.cshlp.org/cgi/content/abstract/2009/12/pdb.top65http://cshprotocols.cshlp.org/content/2009/12/pdb.top65.full`.

Philipp J Keller, Annette D Schmidt, Joachim Wittbrodt, and Ernst H K Stelzer. Reconstruction of zebrafish early embryonic development by scanned light sheet microscopy. *Science (New York, N.Y.)*, 322(5904):1065–9, November 2008. ISSN 1095-9203. doi: 10.1126/science.1162493. URL `http://www.ncbi.nlm.nih.gov/pubmed/18845710http://www.sciencemag.org/content/322/5904/1065.longhttp://www.embl.de/digitalembryo/fish.html`.

Erik Meijering, Oleh Dzyubachyk, and Ihor Smal. *Methods for cell and particle tracking.*, volume 504. Elsevier Inc., 1 edition, January 2012. ISBN 9780123918574. doi: 10.1016/B978-0-12-391857-4.00009-4. URL http://www.ncbi.nlm.nih.gov/pubmed/22264535.

M J Saxton and K Jacobson. Single-particle tracking: applications to membrane dynamics. *Annual review of biophysics and biomolecular structure*, 26:373–99, January 1997. ISSN 1056-8700. doi: 10.1146/annurev.biophys.26.1.373. URL http://www.ncbi.nlm.nih.gov/pubmed/9241424.

I F Sbalzarini and P Koumoutsakos. Feature point tracking and trajectory analysis for video imaging in cell biology. *Journal of structural biology*, 151(2):182–95, August 2005. ISSN 1047-8477. doi: 10.1016/j.jsb.2005.06.002. URL http://www.ncbi.nlm.nih.gov/pubmed/16043363.

Ivo F Sbalzarini. A MATLAB toolbox for virus particle tracking ICoS Technical Report Notice of Copyright. *Technology*, pages 1–16, 2006. URL https://www1.ethz.ch/mosaic/research/docs/Sbalzarini_SPT.pdf.

P Vallotton, a Ponti, C M Waterman-Storer, E D Salmon, and G Danuser. Recovery, visualization, and analysis of actin and tubulin polymer flow in live cells: a fluorescent speckle microscopy study. *Biophysical journal*, 85(2):1289–306, August 2003. ISSN 0006-3495. doi: 10.1016/S0006-3495(03)74564-0. URL http://linkinghub.elsevier.com/retrieve/pii/S0006349503745640http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1303246\&tool=pmcentrez\&rendertype=abstract.

Walther Vogt. Gestaltungsanalyse am Amphibienkeim mit örtlicher Vitalfärbung. Vorwort über Wege und Ziele. *Wilhelm Roux' Archiv für Entwicklungsmechanik der Organismen*, 106(1-4):542–610, December 1925. ISSN 0949-944X. doi: 10.1007/BF02079545. URL http://www.springerlink.com/index/10.1007/BF02079545.