



Universiteit
Leiden
The Netherlands

Image Analysis and Platform Development for Automated Phenotyping in Cytomics

Yan, K.

Citation

Yan, K. (2013, November 27). *Image Analysis and Platform Development for Automated Phenotyping in Cytomics*. Retrieved from <https://hdl.handle.net/1887/22550>

Version: Corrected Publisher's Version

License: [Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden](#)

Downloaded from: <https://hdl.handle.net/1887/22550>

Note: To cite this publication please use the final published version (if applicable).

Cover Page



Universiteit Leiden



The handle <http://hdl.handle.net/1887/22550> holds various files of this Leiden University dissertation.

Author: Yan, Kuan

Title: Image analysis and platform development for automated phenotyping in cytomics

Issue Date: 2013-11-27

Chapter 3

Robust Object Tracking for Cytomics

This chapter is based on the following publications

Yan, K., LeDévédec, S., van de Water, B., Verbeek, F.J. (2009) "Cell Tracking and Data Analysis of in vitro Tumour Cells from Time-Lapse Image Sequences". *In Proc. of International Conference on Computer Vision Theory and Application (VISAPP2009)*.

Yan, K., Le Dévédec, S., Van de Water, B., & Verbeek, F. J., "Automated Analysis of Matrix Adhesion Dynamics in Migrating Tumor Cells", (in preparation)

Chapter Summary

Object tracking or video tracking is the procedure of following moving objects over consecutive frames in a video. The main principle of a tracking algorithm is to associate target objects from consecutive frames based on given linkage criteria such as minimum shape change or motion model. When performing dynamic analysis with a live cell HT/HC screen, object tracking is essential to provide phenotypical quantifications on migration behavior. Unlike standard video recording, HT/HC live imaging is based on time-lapse microscopy with a lower temporal-resolution of only one frame for every couple of minutes. In HT/HC live imaging, the quantity, in terms of number of time points, must often be sacrificed to guarantee the image quality. As a result, the selection of a robust object tracking algorithm is important in the dynamic analysis HT/HC live imaging.

This chapter introduces four tracking algorithms generally applied in HT/HC live cell screens using time-lapse imaging. These algorithms are divided into **confidence measurement based** and **motion-model based**. The confidence measurement based tracking algorithms include the **blob tracking** algorithm and the **kernel density estimation (KDE) with mean shift** tracking algorithm. These assume, regardless of migration, a minimum shape change only occurs between consecutive objects. Thus, often a shape model (KDE) or an overlap measurement (blob) are employed as linkage criteria. The **motion model** tracking algorithm assumes that the object is moving in a quantifiable probability model and by knowing the previous location of the object it is possible to predict the next position of the object. Typical motion model tracking algorithms are the **particle filter** tracking algorithm employing Brownian motion model and the **energy driven linear (EDL) model** tracking algorithm employing linear motion model. The confidence measurement tracking algorithms should not be mixed with the motion model tracking algorithms since the former is an analysis based on shape similarity while the latter is an analysis relying on motion patterns.

3.1. Introduction

Object tracking or video tracking is the image analysis procedure concerned with the linkage of objects. Here we define a video is a collection of frame captured at a temporal-resolution higher than 25 frame/second while a time-lapse image sequence is a type of video captured at a temporal-resolution lower than 25 frame/second. A HT/HC live cell screens in Cytomics often produces time-lapse image sequence instead of video.

Unlike object tracking with standard video, object tracking with time-lapse image sequence faces several complexities; amongst which the velocity-to-temporal-resolution ratio (VTR) is the most essential one. We define the VTR as the ratio between object's real velocity and temporal resolution of the video. A high VTR suggests there are more variations between objects from consecutive frames. In a high VTR image sequence, the between-frame association of the recognized object can be difficult. Moreover, motion of objects in live cell HT/HC screen are often the result of nonlinear deformation [91][92] that cannot be described by a rigid model [93][94][95]. The deformations introduce more morphological variation between objects. To guarantee the reliability of the tracking, the VTR of a time-lapse image sequence must exceed a minimum threshold that captures major morphological changes in objects.

There are generally two sides to object tracking with time-lapse image sequences: (1) the recognition of the relevant objects and (2) the between-frame association of the recognized object. Recognition of the relevant objects is accomplished using image segmentation (cf. Ch. 2). It is a laborious process due to the large volume of image data and its complexity is often determined by the robustness of image segmentation techniques. The tracking algorithms are based on, but not limited to, the following two properties:

1. confidence measurement [9][10]
2. motion models [31][17]

The property (1) includes shape similarity measurements such as kernel density estimation [9][96] while the property (2) includes probability functions such as Brownian motion model [97][98]. In general, confidence measurement based tracking algorithm such as blob and mean shift tracking algorithms assume that a minimum shape change will only occur between consecutive objects while the motion model based tracking algorithms, such as particle filter tracking and energy driven tracking, assume that the next location can be predicted from all previous locations of the object. Recent studies show that further optimization based on tree diagrams [99] can improve the tracking decision.

Tracking algorithms frequently used in HT/HC live cell screen are blob tracking, mean shift tracking, active contour tracking, and particle filter tracking:

1. The blob tracking is a feature point based tracking algorithm that constructs object linkage using confidence estimation. It is a computationally cheap tracking algorithm that works well real-time tracking problems.
2. The mean shift tracking is a model based tracking algorithm based on confidence estimation. Unlike blob tracking, the mean shift tracking algorithm recursively shifts the initial model to the most likely region in the consecutive frames.

3. Active contour tracking is a model based tracking algorithm that propagates the initial contour to each consecutive frame. It is a computationally intensive algorithm that requires human interference.
4. The particle filter tracking is a feature point based tracking algorithm similar to blob tracking.

In this chapter, all tracking algorithms i.e. **blob tracking**, **KDE mean shift tracking**, **particle filter tracking**, **energy driven linear model tracking**, as well as **active contour tracking** will be explained first. Subsequently, we perform a systematic estimation of tracking efficiency within the application domain of HT/HC studies using manually produced ground truth data sets. Finally, we will demonstrate the motivation behind selection of tracking algorithms in different combinations of imaging and research in cell biology.

Blob Tracking Algorithm

Blob tracking [100][101][102][97][33] is a straightforward tracking algorithm that links objects by confidence measurement such as measuring distance [103][104] or size changes between the target object and candidate object from consecutive frames (cf. Figure 3-1). However, as it is based on confidence measurement concepts, blob tracking can only track object with a mostly rigid body transformation[103].

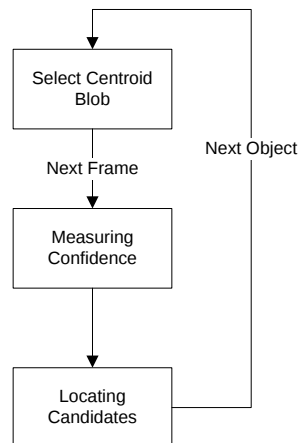


Figure 3-1 simple workflow of the blob tracking algorithm

Mean Shift Based Tracking Algorithm

The mean shift algorithm [105] is considered a real world application of an model based localization approach. It is a robust tracking solution [99][106][107][9][108][15] to associate object linkage by localizing an initial model in consecutive frames. For one n-frame video, the mean shift algorithm starts by converting the initial object into multiparametric density models and recursively update the mean shift factor based on a local density in the consecutive frames until a stationary location is reached. Finally, it associates the initial object with the candidates closest to the station location (cf. Figure 3-2). A kernel based mean shift tracking consists of two steps[105]:

1. Non-parametric density estimation from an initial model.
2. Steepest descent to locate the local maximum in a gradient space of density estimations given an initial model.

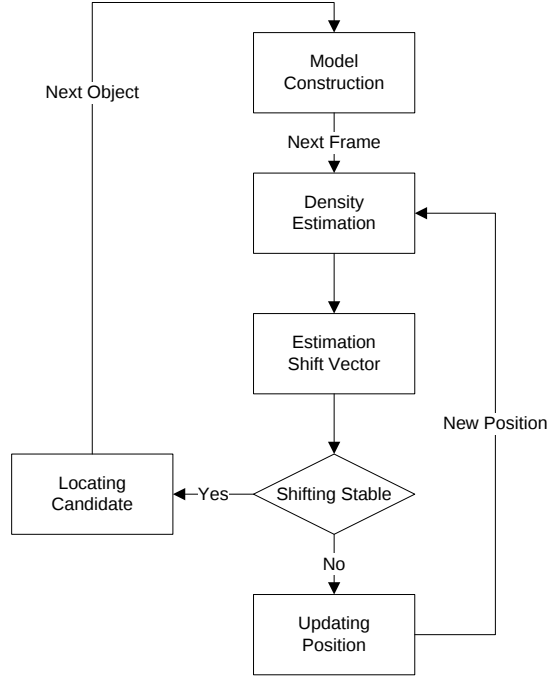


Figure 3-2 simple workflow of the mean shift based tracking algorithm

Each trajectory begins with objects in one frame. These objects are converted into initial model defined in a multidimensional feature space. These dimensions include (1) the x -coordinate of a binary mask of an object, (2) the y -coordinate of a binary mask of an object, (3) the intensity value at each pixel (x, y) . Given n data points x_n in the d -dimensional space R^d , the kernel density estimator with kernel function $K(x)$ (cf. Equation 3-2) and window bandwidth h , can be expressed as:

$$\hat{f}_{h,k}(x) = \frac{c_{k,d}}{nh^d} \sum_{i=1}^n K\left(\left\|\frac{x-x_i}{h}\right\|^2\right) \quad \text{Equation 3-1}$$

A Gaussian kernel is used as a radial symmetric kernel, expressed as:

$$K_N(x) = (2\pi)^{-\frac{d}{2}} |\Sigma|^{-\frac{1}{2}} e^{-\frac{1}{2} \Sigma^{-1} \|x\|^2} \quad \text{Equation 3-2}$$

Subsequently, the mean shift estimation is completed by steepest descent through iterative computation of:

- the mean shift vector $m(k^{th})$
- the shifted model by $x^{k+1} = x^k + m(k^{th})$

The steepest descent requires estimation of the gradient space $g(x) = -k'(x)$, where the

mean shift vector $m(k^{th})$ is calculated by $\frac{\sum_{i=1}^n x_i g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)}{\sum_{i=1}^n g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)} - x$. Because of the shape change

(deformation) of objects, the steepest descent does not necessarily converge at the centre of mass of the true candidate. We chose the object closest to the stationary point, at which the magnitude of (k^{th}) is closest to zero, as the probable candidate.

KDE mean shift tracking cannot provide a precise localization of the candidate position. However, it is more robust and flexible since the multiparametric density model can be derived

from any available information. As a result, the mean shift based algorithm does not necessarily rely on the relative position of candidates. Therefore, it is a good solution for tracking motile objects in time-lapse image sequences of low temporal-resolution.

Particle Filter Based Tracking Algorithm

Particle filter based tracking [36][65][109] is a lineage linkage based tracking method frequently employed in the study of random particle activity in physics [110]. This is a category of tracking algorithms that relies on a series of observations containing stochastic variation over the temporal dimension so as to produce a recursive statistical approximation; i.e., the spreading function of the underlying system states (cf. Figure 3-3). To further improve tracking accuracy, particle filter based tracking often includes linking strategies such as graph-based optimization [98] or Bayesian recursive estimation [110]. Particle filter based algorithms have been used in cell biology tracking problems like tracking random cell migration at low magnification [103].

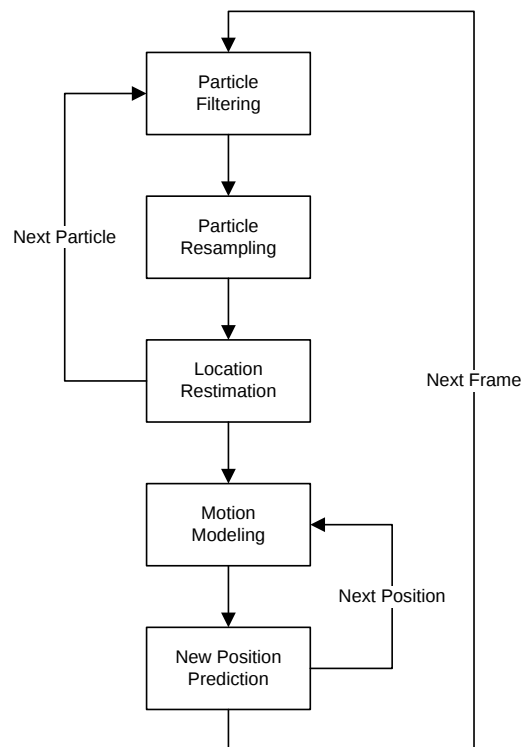


Figure 3-3 Simple workflow of particle filter based tracking

Particle filter tracking method using Bayesian recursive estimation assumes that input x_k and the observations y_k can be modeled in this form:

- $x_0, x_1, \dots, x_m$ is a first order Markov process such that $x_k | x_{k-1} \rightarrow p_{x_k | x_{k-1}}(x_k | x_{k-1})$ at state k and with an initial distribution $p(x_0)$.
- The observation $y = \{y_0, y_1, \dots, y_n\}$ are conditionally independent provided that $x_0, x_1, \dots, x_m$ are known. So each y_k only depends on x_k , $y_k | x_k \rightarrow p_{y_k | x_k}(y_k | x_k)$.

The linkage is constructed based on recursively updating of the posterior distribution over the current state x_t given all observations $Y = \{Y_0, Y_1, \dots, Y_p\}$ up to a discrete frame number t as follows.

$$p(x_t|Y_t) = kp(x_t|y_t) \int_{x_{t-1}} p(x_t|x_{t-1})p(x_{t-1}|Y_{t-1}) \quad \text{Equation 3-3}$$

where the likelihood $p(x_t|y_t)$ at time t expresses the measurement model and $p(x_t|x_{t-1})$ is the motion model. The posterior probability $p(x_{t-1}|Y_{t-1})$ is approximated recursively as a set of weights of N samples $\{x_{t-1}^{(r)}, \pi_{t-1}^{(r)}\}_{r=1}^N$, where $\pi_{t-1}^{(r)}$ is the weight for the particle $x_{t-1}^{(r)}$. Using Monte Carlo approximation, Equation 3-3 can be calculated using Equation 3-4, whereas each particle is scored according to the approximation (cf. Equation 3-4).

$$p(x_t|Y_t) \approx kp(x_t|y_t) \sum_r \pi_{t-1}^{(r)} p(x_t|x_{t-1}^{(r)}) \quad \text{Equation 3-4}$$

Energy Driven Linear Tracking Algorithm

The energy driven linear tracking algorithm (EDL) [16] is a particle filter tracking algorithm that we designed for our tracking problems in subcellular structures known as the matrix adhesion (MA). The EDL is based on empirical observation of MA dynamics [36][111][112] from which MAs are believed to move in a linear fashion along stress fibers. The EDL tracking algorithm consists of the following steps:

1. Multivariate Gaussian model construction
2. Density estimation
3. Pairwise linkage
4. Trajectory construction

The complete workflow is illustrated in the Figure 3-4. We will first construct a probability model that describes the pseudo-motion behavior. Next, we denote X_n as the pixels set of the binary mask of one object at time point n .

For one X_n , the pixel coordinates of its binary mask are stored as a $(N \times 2)$ matrix. Then the center of mass set $\mu_n = E(X_n)$ is first calculated and next from X_n and μ_n , the covariance matrix S_n is calculated as:

$$S_n = \frac{(X_n - \mu_n)(X_n - \mu_n)^T}{N - 1} \quad \text{Equation 3-5}$$

Given object μ_n and S_n , the algorithm further calculates the new center of mass μ_n' by $\mu_n' = \mu_n + V_s$, where the shifting vector V_s is calculated as $V_s = \mu_n - \mu_{n-1}$, where the μ_{n-1} is the mass center of X_{n-1} . For each X_n , an updated multivariate Gaussian model $f(X, \mu_n', S_n)$ is constructed based on S_n and μ_n' as:

$$f(X, \mu_n', S_n) = \frac{1}{(2\pi)^{\frac{k}{2}} \det(S_n)^{\frac{1}{2}}} \exp\left(-\frac{1}{2} \frac{(X - \mu_n')^T (X - \mu_n')}{S_n}\right) \quad \text{Equation 3-6}$$

, where the k is the column rank of X_n ; which in this case $k = 2$ since there are two columns.

Next for all X_{n+1} within the maximum searching radius r_{max} , the average per-pixel probability of X_{n+1} is calculated as:

$$p(X_{n+1}) = \frac{1}{N_{n+1}} \sum f(X_{n+1}, \mu_n', S_n) \quad \text{Equation 3-7}$$

The average per-pixel probability p of X_{n+1} is employed as the score of the linkage. Figure 3-5 illustrates a sample result of the score table. Instead of looking at a local optimum of linkage, the pairwise linkage intends to construct a linkage of global optima between time point n and $n+1$. If X_n can be linked to two successors, the algorithm always selects the pair with the larger per-pixel probability given $f(X, \mu_n', S_n)$.

Finally, the algorithm searches through all pairs and constructs a trajectory. If X_n has no valid successor, the trajectory will be terminated. For this we use the following termination function F :

$$F(X_{n+1}) = \begin{cases} p(X_{n+1}) & \geq p_{min} \\ \|\mu_n - \mu_{n+1}\| & \leq r_{max} \end{cases} \quad \text{Equation 3-8}$$

, where the termination criterion r_{max} is an user-defined maximum search radius and the termination criterion p_{min} is an user-defined minimum per-pixel probability.

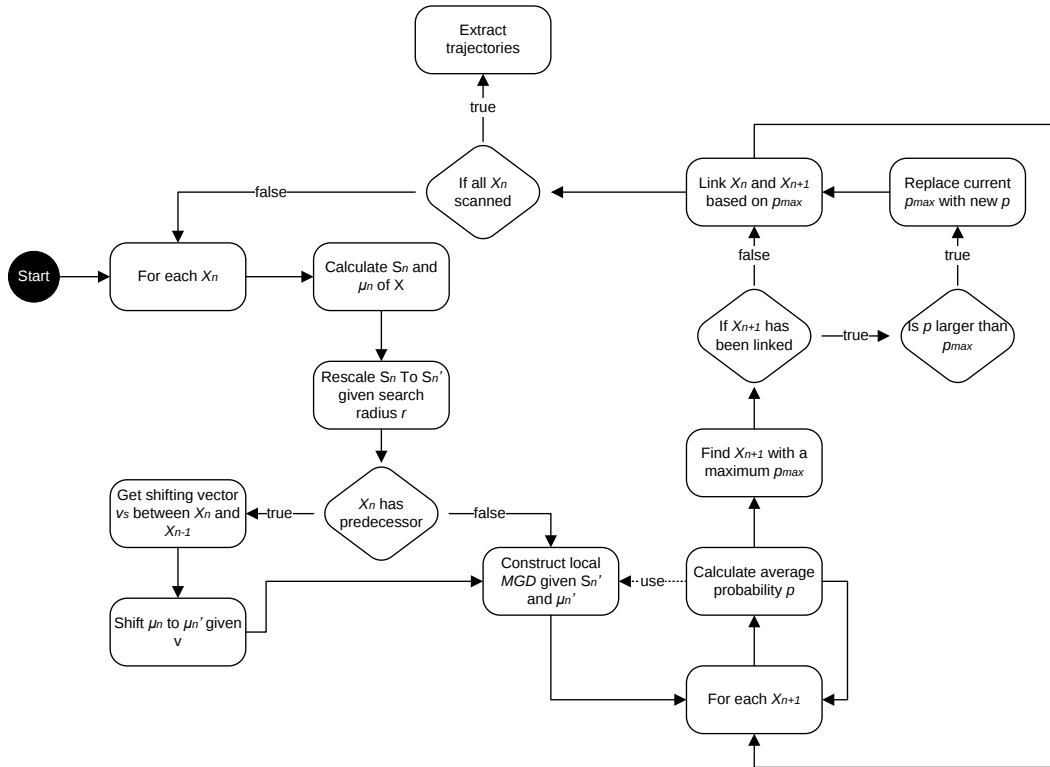


Figure 3-4 workflow of energy driven linear tracking

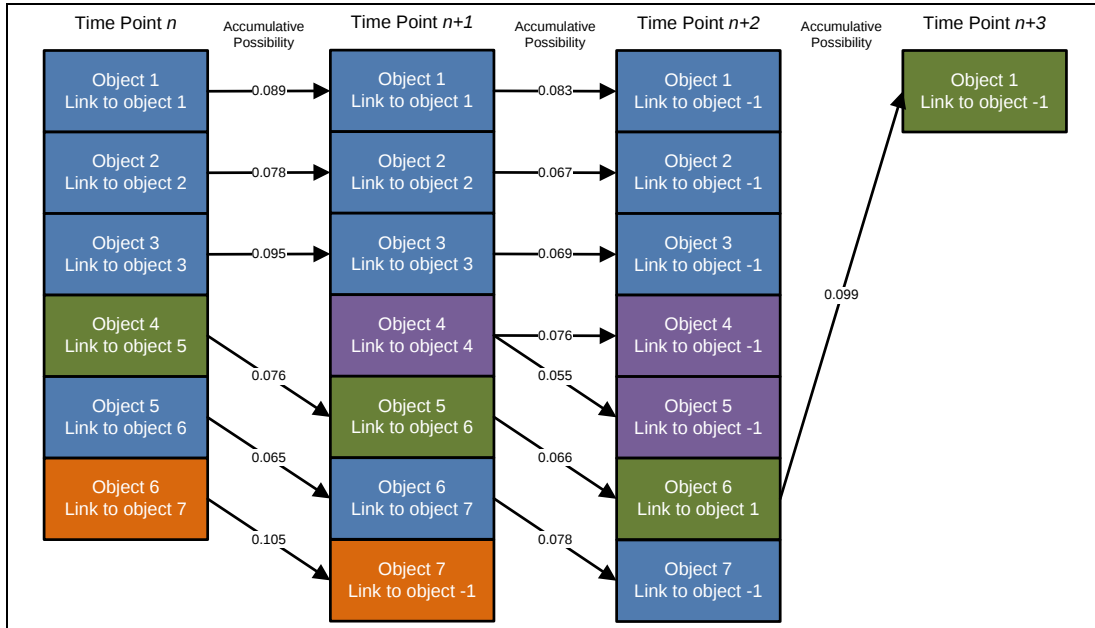


Figure 3-5 score table and pairwise linkage schema

Active Contour Tracking Algorithm

The active contour algorithm [101][103][45] is a top-down tracking algorithm combining both segmentation and tracking. From an initial model, active contour based tracking attempts to minimize an objective function associated with the current model in a recursive fashion. For an n -frame video, the active contour algorithm first adapts the initial contour for the object in frame i and then uses the adapted contour as the initial contour for frame $i+1$ until it reaches frame n . Such recursive propagation of initial model not only provides a self-adaptive segmentation procedure, but also an association between objects in consecutive frames. The accuracy and robustness of active contour based tracking is subject to the choice of energy definition and local search mechanism; these must be changed for different sets of image sequences. Moreover, active contour tracking always requires a manual initialization. Therefore, the active contour based tracking is less applicable in high-throughput analysis. We included a demo tracking using active contour tracking, but did not extend this to a large scale test.

We selected an open-source implementation for each tracking algorithm category since these open-source implementations have already been used and optimized in the application domain of HT/HC studies. Using open-source implementations provides an independent starting point for the assessment of the performance of each algorithm.

- Blob tracking → MTrack2 plug-in in Fiji [64]
- Mean shift tracking → KDE mean shift plug-in in ImageJ [9]
- Particle filter tracking → particle detector and tracking plug-in in Fiji [109]
- Active contour tracking algorithm → AB snake in ImageJ [65]

These algorithms are frequently employed in HT/HC studies. The efficiency of these tracking algorithms has not yet been assessed. In the next section, we perform a systematic quantification and characterization of the tracking performance of each algorithm in the application domain of HT/HC live cell screens.

3.2. Performance Study

The tracking efficiency of each algorithm is measured from a ground truth time-lapse image sequence containing manually segmented and tracked objects. Using manually segmented objects (cf. Figure 3-6c), we can assess the tracking accuracy of each algorithm without considering possible segmentation errors. Moreover, since each algorithm uses different information for linkage calculation, the manually segmented objects provide a mutual starting point by restrict tracking to be performed only with the manually segmented objects.

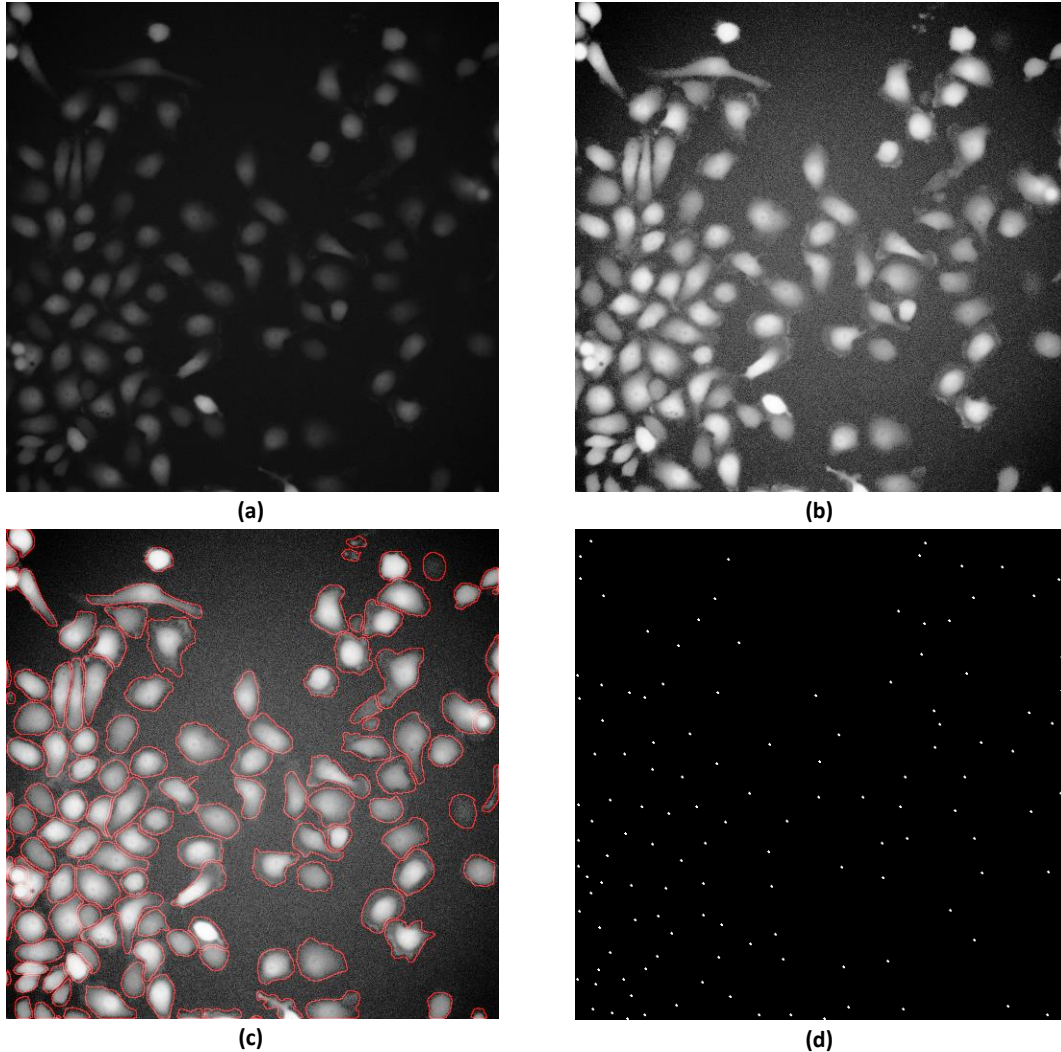


Figure 3-6 (a) raw image from the HT/HC screen of random cell migration, (b) intensity equalized version of raw image (a), (c) manually segmented and tracking cells, (d) mass center of each object

To assess tracking efficiency, here we introduce an assessment metrics consisting of five elements [113]: (1) true positive tracking, (2) false positive tracking, (3) global detection rate, (4) global fragmentation rate and (5) final score. The procedure is depicted in Figure 3-7 and described as following:

Step 1 Ground Truth Production

Observer is required to produce binary masks and trajectories of dynamic objects based on given time-lapse image sequence.

Step 2 Tracking with Prior Object Annotation

The manually segmented binary mask (cf. Figure 3-6c) will be used by the blob and mean shift tracking algorithms. The manually segmented binary mask is converted into centers of mass for particle detection and tracking (cf. Figure 3-6d). In theory, such design should guarantee that all tracking algorithms are tracking the same set of objects.

Step 3 Converting Tracking Output to Mutual Ground

Each trajectory is converted into a multi-dimensional array (cf. Table 3-1) in which the *path#* is the unique index of each trajectory, the *frame#* is the frame index, the (*MC_X*, *MC_Y*) is the (*x*, *y*) coordinate of the center of mass the object. Given one trajectory, the extraction of corresponding object masks is illustrated in Figure 3-9. Since the different algorithm implementations produce various formats of output to describe trajectory information, the only possible way to compare the tracked trajectories from different implementation is to map trajectories back to the objects.

Step 4 True Positive Tracking (TP) and False Positive Tracking (FP)

Using state definition in Figure 3-8, the true positive tracking and false positive tracking are calculated for each tracking algorithm.

Step 5 Global Detection Rate (GDR)

Using the result from true positive tracking and false positive tracking, each path tracked is assigned to one ground truth path. From the global coverage, the global detection rate is calculated.

Step 6 Global Fragmentation Rate (GFR)

Using fragmentation definition in Figure 3-8, the global fragmentation rate is calculated for each tracking algorithm.

Step 7 Final Tracking Score

The final tracking score is derived from TP, FP, GDR and GFR (cf. Equation 3-16).

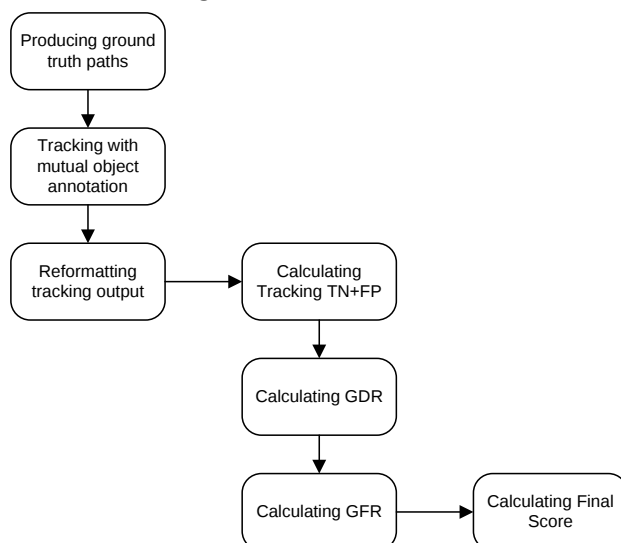


Figure 3-7 Workflow of the tracking efficiency estimation

Table 3-1 Sample of trajectory table

Path#	Frame#	MC_X	MC_Y
0	0	253.2	278.9
0	1	260.9	283.4
0	2	259.9	288.8
0	3	255.3	287.9
0	4	250.5	285.3
0	5	246.8	285.3

We illustrate the tracking efficiency of the algorithms against two types of image modalities:

1. HT/HC screen of random cell migration using epifluorescence microscopy
2. HC analysis of matrix adhesion (MA) dynamics using TRIF microscopy

The manual segmentation and tracking of each object or observation, either cell or MA, are both accomplished by biologists (observers). Image quality was first enhanced (cf. Figure 3-6b) before being processed by each observer. During the observation, each observer starts with one cell and continuously draws the outline of the same cell along the entire sequence (cf. Figure 3-6c). When cell division occurs, the observer will track both daughter cells, separately, starting from the mother cell. In that particular case, there will be two manually tracked cells that have a partial identical trajectory. In the next section, the definition of the tracking efficiency metrics will be addressed.

3.2.1. Tracking Efficiency Metrics

True Positive Tracking (TP) and False Positive Tracking (FP)

First, the true positive tracking and false positive tracking are defined for the tracking results. When comparing the tracked path to the ground truth path (cf. Figure 3-9), there are potentially three states for each time point in the trajectory (cf. Figure 3-8). The ‘match’ state is the good scenario at one time point in which objects from two paths are identical. The ‘mismatch’ state is the state at one time point in which objects from two paths are not identical. The ‘no match’ state is the state at one time point in which there is no corresponding object presented in either the tracked path or the ground truth path. From these three states, we define the true positive and true negative of the tracking as following:

1. The true positive tracking (cf. Equation 3-9) is measured by the number of objects in a trajectory with a ‘match’ state C_{match} divided by the length of the ground truth path L_{gt} .
2. The false positive tracking (cf. Equation 3-10) is measured by the number of objects in a trajectory with a ‘no match’ state C_{no_match} divided by the length of the ground truth path plus the length of the path being tracked $L_{gt} + L_{tp}$.
3. The true negative (TN) is calculated by $TN = 1 - FP$.

$$TP = \frac{C_{match}}{L_{gt}} \quad \text{Equation 3-9}$$

$$FP = \frac{C_{no_match}}{L_{gt} + L_{tp}} \quad \text{Equation 3-10}$$

Global Detection Rate (GDR)

The global detection rate (cf. Equation 3-11) in tracking is a quantification of the success rate that each tracking algorithm can recognize all available paths. It is an indicator of the sensitivity of tracking algorithm to detect potential paths. A good tracking algorithm should not only fully recover a certain path but also recover all possible paths. Given ground truth path set T_{gt} and tracked path T_{tp} , and intersection T_{inter} between T_{gt} and T_{tp} , the global detection rate is calculated from the size of T_{inter} divided by the size of T_{gt} . The intersection T_{inter} the path overlapping calculated from tracking $TP + FP$.

$$GDR = \frac{T_{gt} \cap T_{tp}}{T_{gt}} \quad \text{Equation 3-11}$$

Global Fragmentation Rate (GFR)

The global fragmentation rate in the tracking is the percentage of paths that are only tracked partially or available as fragmented trajectories. It is a supplementary quantification to global detection rate. In contrast to the global detection rate, which is an indicator of overall tracking sensitivity, the fragmentation rate is an indicator of overall tracking specificity. Figure 3-8 illustrates a partially tracked result in which a tracking algorithm produces two path fragments that should belong to one ground truth path.

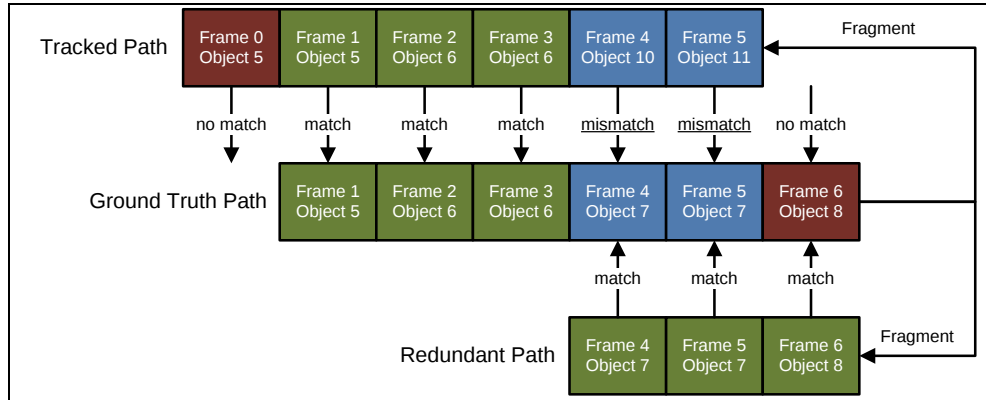


Figure 3-8 three states for each time point in one trajectory, red block: no match, green block: match, blue block: mismatch.

Final Score (FS)

The final score of tracking is the tracking efficiency weighted by the detection rate and the fragmentation rate. It indicates the overall performance of each tracking. The F1-score [87] is calculated given tracking TP and FP (cf. Equation 3-12, Equation 3-13 and Equation 3-14) and weighted using both GDR and GFR (cf. Equation 3-15). Contrary to weighted average, the production based final score enforces a balance amongst all subsystems. The final score is equal to one only if all three subsystems are approaching to one. Unlike a weighted average, if one of the subsystems approaches zero, the whole final score will approach zero.

$$sensitivity = \frac{TP}{TP + FN} \quad \text{Equation 3-12}$$

$$specificity = \frac{TN}{TN + FP} \quad \text{Equation 3-13}$$

$$F1 = 2 \cdot \frac{sensitivity \cdot specificity}{sensitivity + specificity}$$

Equation 3-14

$$FS = GDR \cdot (1 - GFR) \cdot F1$$

Equation 3-15

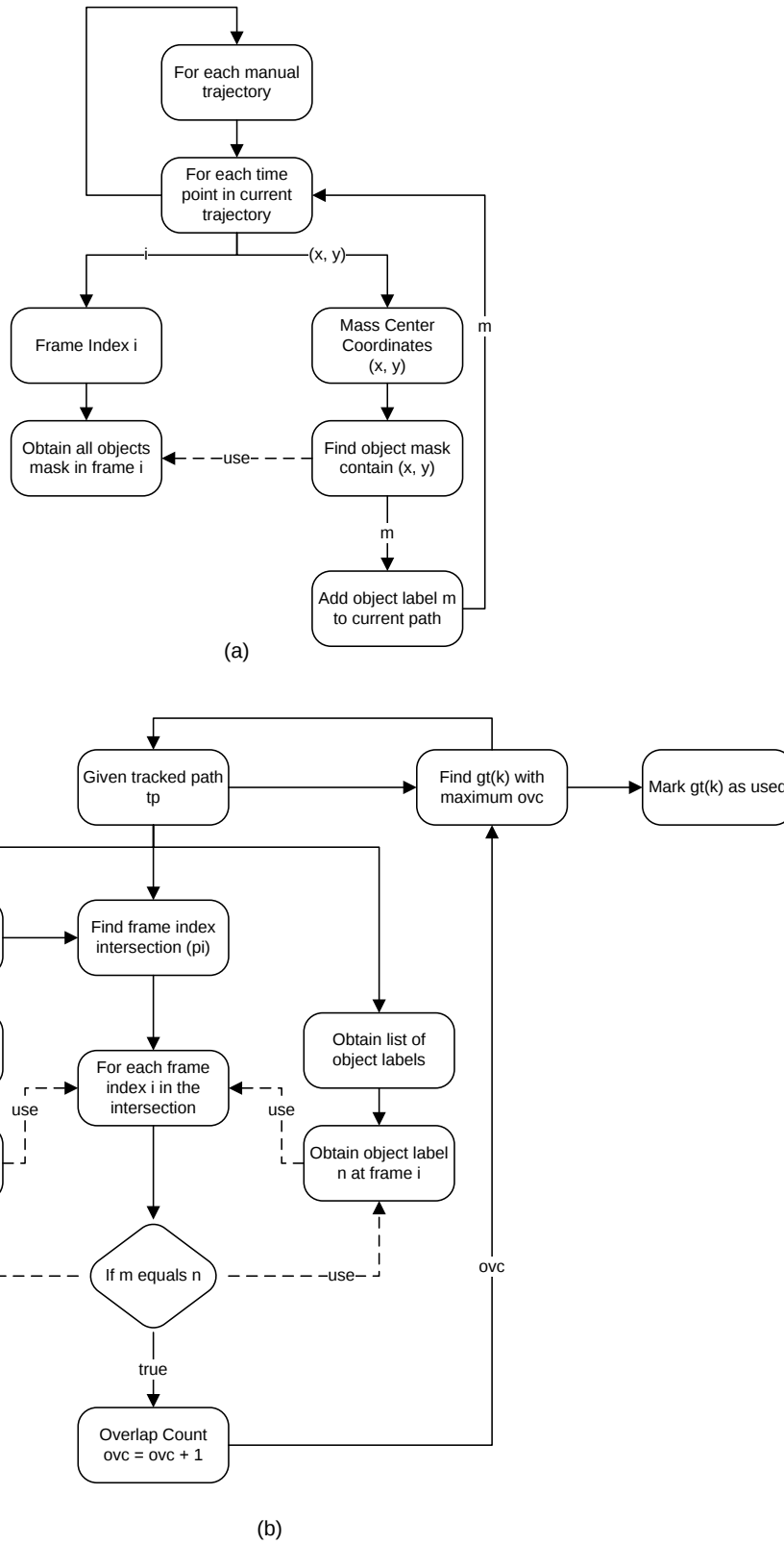


Figure 3-9 (a) extraction of an object mask given one trajectory (b) calculating trajectory overlap

3.2.2. Results of Efficiency Estimation

MTLn3 pGFP Ground Truth

The MTLn3 pGFP is an aggressive breast cancer cell line that is regularly used in studies on migration suppression [15] and growth factor inhibition [114]. This cell line has a motile behavior and being able to track these cells is often considered a major challenge in cell biology. For the experiment, the image acquisition is accomplished using Nikon TE2000 epifluorescence microscope with a 40x lens (NA 0.75) and captured at a fixed temporal resolution of 5 min/frame.

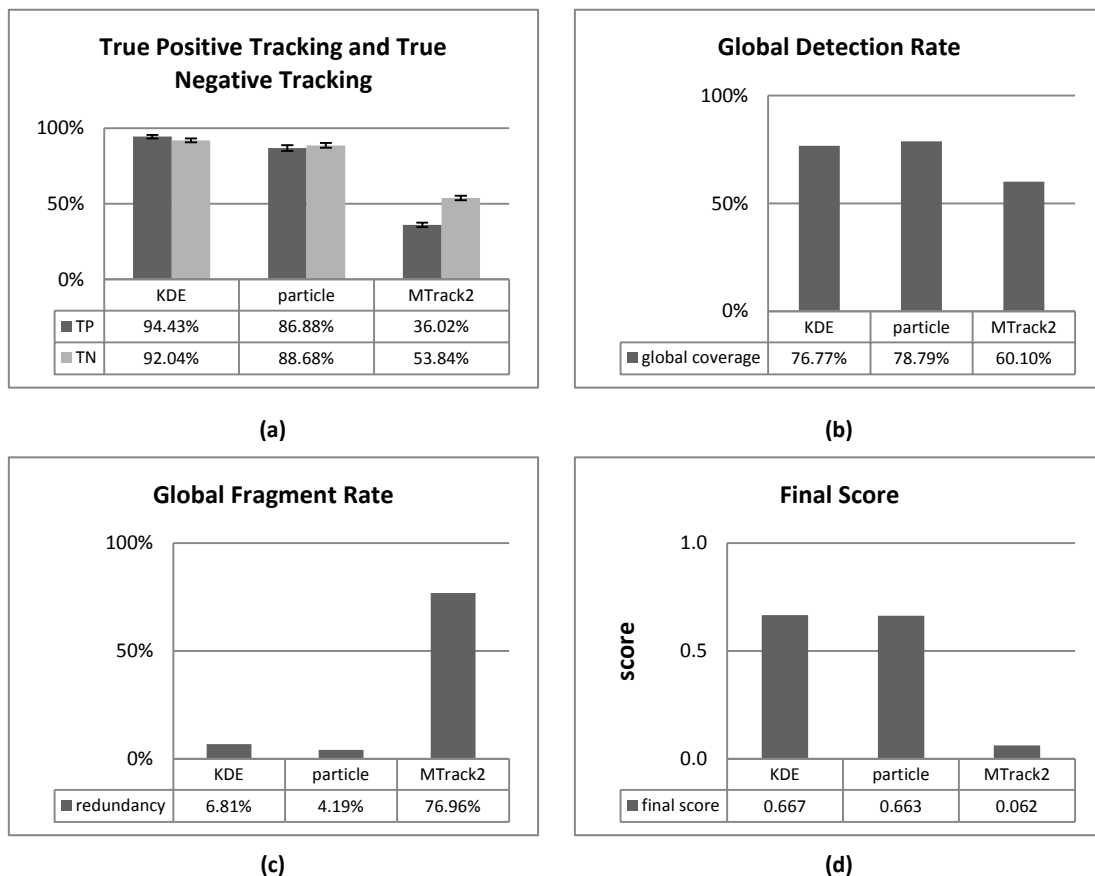


Figure 3-10 efficiency estimation using MTLn3 pGFP test set

The efficiency estimation (cf. Figure 3-10a) of each tracking algorithm shows that the KDE mean shift algorithm yields the highest TP-rate of 94.43% and the highest TN-rate of 92.04%. The particle filter tracking yields a TP-rate of 86.88% and a TN-rate of 88.68%. The MTrack2 yields the lowest TP-rate of 36.02% and the lowest TN of 53.84%. The result shows that KDE mean shift tracking produces the best overall result compared to both particle filter and MTrack2 tracking. The slightly lower performance of the particle filter tracking is mostly due to the underlying assumption of Brownian motion. The Brownian motion model is a popular function to describe random motion of particles; however, we observe that random cell migration does not necessarily follow this pattern. Therefore, particle filter tracking is less efficient compared to KDE mean shift tracking in cell migration. The performance of MTrack2 is mostly due to the variable acceleration in cell migration; a sudden acceleration of cell motility

may cause the MTrack2 to prematurely terminate a track, leading to small fragments of trajectories which, *de facto*, should be merged.

The global detection rate (cf. Figure 3-10b) shows the highest score for particle filter tracking 78.79%; KDE mean shift tracking has slightly lower score of 76.77%. The MTrack2 scores 60.10%. In our test, the results illustrate that the particle filter tracking demonstrates an acceptable sensitivity when detecting potential paths. The performance of KDE mean shift tracking algorithm is similar to the particle tracking algorithm. The MTrack2 ignores nearly half of the potential paths.

The global fragmentation rate (cf. Figure 3-10c) shows that the KDE mean shift tracking has a chance of 6.81% to produce fragmented tracking result. The particle filter tracking shows the lowest fragmentation of 4.19%. Due to its distance based criterion and cells' random acceleration, MTrack2 has the highest chance of performing a fragmented tracking.

The final score (cf. Figure 3-10d), which brings all error estimation together, shows that in general KDE mean shift tracking algorithm and particle filter tracking algorithm both demonstrate robust performance in tracking the MTLn3 pGFP cell line while MTrack2 is clearly underperformed due to a prefixed tracking criterion. Generally, the performance of KDE mean shift tracking algorithm is slightly higher than the particle filter tracking algorithm. Compared to the particle filter tracking algorithm, the KDE mean shift tracking algorithm produces a result with significantly higher TP value and TN value. Since most phenotypical quantifications at the single-cell level are derived per-trajectory instead of whole population, a higher per-trajectory error may lead into wrong quantification while a redundant tracking may only shift the data distribution. Thus, it is preferable to use a tracking algorithm with high per-trajectory tracking accuracy.

MTLn3 pGFP + EGF Ground Truth

The ground truth set contains epidermal growth factor stimulated MTLn3 pGFP cells that demonstrate a significantly higher motility even compared to MTLn3 pGFP control cells. The cells are moving approximately five-times faster than MTLn3 pGFP control cells and twenty-times compared to other cell lines. The current image acquisition is accomplished using the Nikon TE2000 epifluorescence mode with a 40x lens (NA 0.75) and captured at a fixed temporal resolution of 5 min/frame. The relatively high motility requires a higher temporal resolution than the actual resolution used. Consequently, manual segmentation and tracking of MTLn3 pGFP +EGF is considerably more difficult.

The tracking error (cf. Figure 3-11a) shows that the tracking efficiency of both the KDE mean shift and the particle filter tracking algorithm is significantly decreased while the performance of MTrack2 is significantly increased. The major reason is due to that cells may move out of the field of view; then the MTrack2 will terminate the tracking while the other two algorithms may attempt to resume the tracking procedure. This also leads to the lower global detection rate of KDE mean shift tracking algorithm (cf. Figure 3-11b) since KDE mean shift intends to combine two ground truth paths into one tracked path when one ground truth path escapes the image frame. Moreover, although the early termination mechanism of MTrack2 may improve

tracking accuracy in MTLn3 pEGFP +EGF, it receives more penalties from the GFR (cf. Figure 3-11c). The final score of MTLn3 pEGFP +EGF (cf. Figure 3-11d) shows a similar tendency compared to the results of MTLn3 pEGFP.

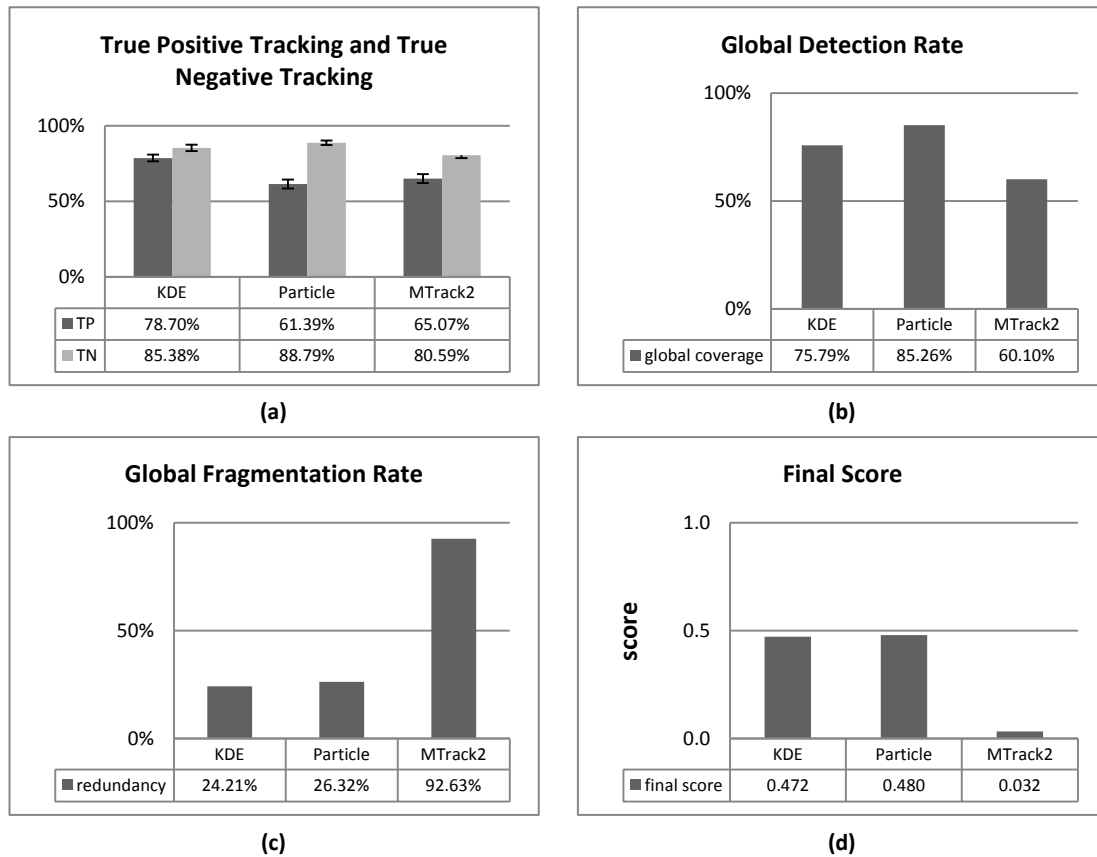


Figure 3-11 efficient estimation using MTLn3 pEGFP+EGF test set

It may be that the KDE mean shift tracking provides a more accurate per-trajectory tracking. However, as the method lacks an early-termination mechanism and escaping detection in the implementation used for testing, its global performance is slightly lower than particle filter tracking. The particle filter tracking algorithm produces a result containing considerable amount of per-trajectory errors. In HT/HC study, it is preferred to have a low per-trajectory error over a low global error since most phenotypical quantifications are derived from each trajectory and not from the whole population. The MTrack2 produces low quality results due to highly aggressive cell behavior and non-linear shape deformation, and resulting in the least desirable outcome. However, from time-lapse sequence with better temporal resolution and less motile cells, it is still a computation inexpensive solution. Moreover, it is foreseeable that with both early termination mechanism and escaping detection implemented, the performance of KDE mean shift tracking algorithm should be higher than both particle filter tracing and MTrack2.

Matrix Adhesion Ground Truth

This ground truth set contains dynamic matrix adhesion (MA) [16][36]. The analysis of MA dynamics is an essential part of cell migration study (cf. Ch. 5). Compared to cell tracking, MAs have a simpler and monotonous morphology. There is only a limited amount of information

that can be used to distinguish two MAs. It complicates the tracking procedure since morphological heterogeneity between potential candidates is reduced, therefore leading to a higher false positive rate when building object linkages.

Based on per-trajectory error estimation (cf. Figure 3-12a), it shows that energy-driven linear tracking algorithm has lower per-trajectory error compared to particle filter and MTrack2. In general, all three tracking algorithms can produce acceptable per-trajectory tracking accuracies. The slightly lower performance of particle filter tracking is mainly due to the assumption of Brownian motion model whereas of a linear motion model better fits the nature of MA dynamics [36].

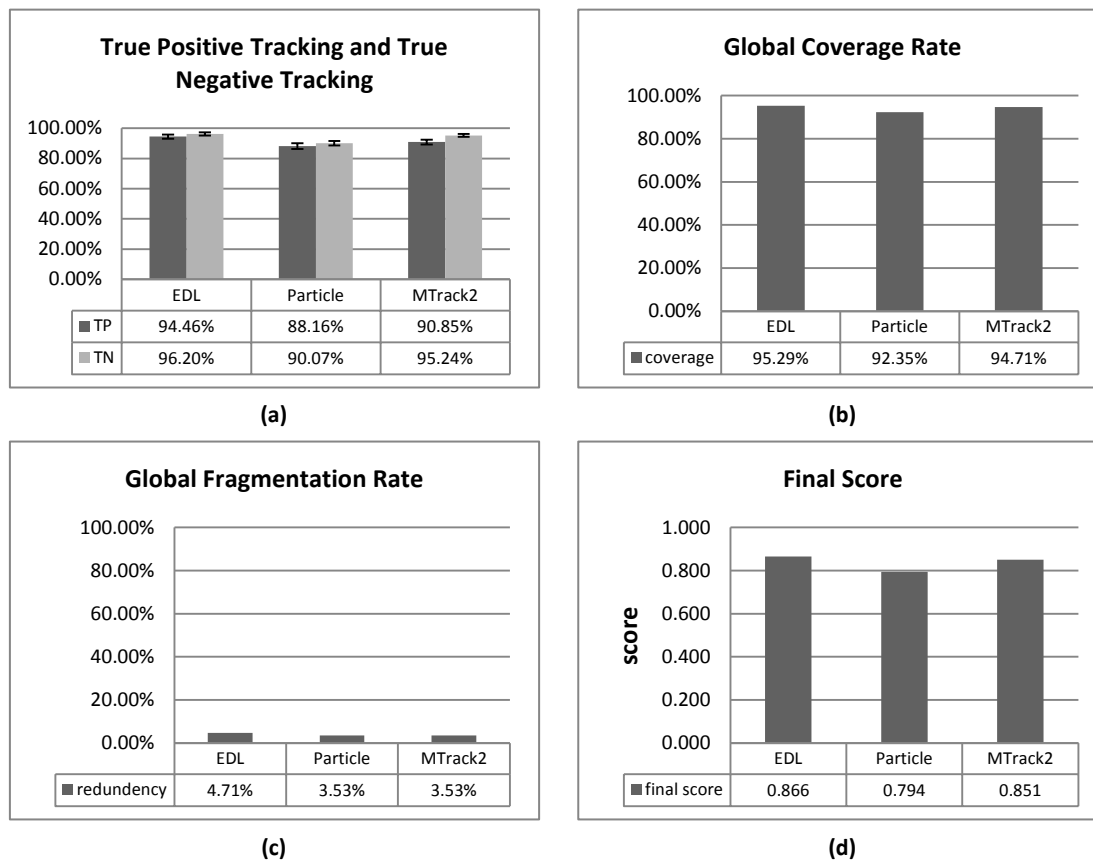


Figure 3-12 efficient estimation using matrix adhesion test set

The global coverage rate (cf. Figure 3-12b) shows that the energy-driven linear tracking algorithm has the highest global coverage of 95.29%. The difference between the three algorithms, however are overall the same.

The global fragmentation rate (cf. Figure 3-12c) shows that the energy-driven linear tracking has a result of lower quality whereas the particle filter and MTrack2 are slightly better. The result suggests that tracking result of EDL may contain more early-terminated trajectories compared to the particle filter and MTrack2 tracking.

The overall performance (cf. Figure 3-12d) shows that the EDL has the best score compared to the particle filter and the MTrack2 tracking algorithm. However, the performance of MTrack2

shows similar performance as EDL. We conclude that both EDL and MTrack2 use all available information from binary mask and intensity landscape while particle tracking must rely on Brownian motion model which seems to be a false assumption in this particular tracking problem.

3.2.3. Temporal Resolution Variance

Temporal resolution is a measurement of video sampling rate. The ratio between temporal resolution and object velocity plays a crucial role in video tracking. A higher velocity-to-temporal-resolution ratio (VTR) suggests that intermediate stage of object motion may be overlooked. As a result, a higher VTR means more variations and fewer similarities between objects in consecutive frames. Here we propose the **Consecutive Displacement Rate (CDR)** measurement of the VTR. The CDR is considered as a numeric indicator of tracking complexity.

Consecutive Displacement Rate

The consecutive displacement rate (CDR) is the pixel-based intensity-weighted difference between consecutive objects. Given consecutive objects A and object B, the CDR is expressed as:

$$CDR = \frac{\overline{A \cap B}}{A \cup B}$$

Equation 3-16

From the algorithm descriptions (cf. §3.1) we can conclude that all tracking algorithms rely on certain similarity criteria when constructing a linkage between consecutive objects. A high CDR suggests that two objects share less overlap, thus, it suggests a potential shape change or significant position shifting. A significant position shifting may introduce an error to the tracking algorithm that relies on relative position such as blob tracking and particle filter tracking while a potential shape change causes a problem to the algorithm such as KDE mean shift tracking. Therefore, a good time-lapse image data set should express the lowest CDR as possible.

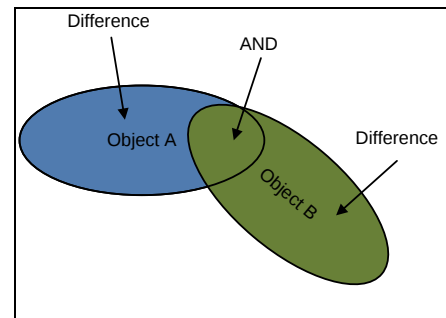
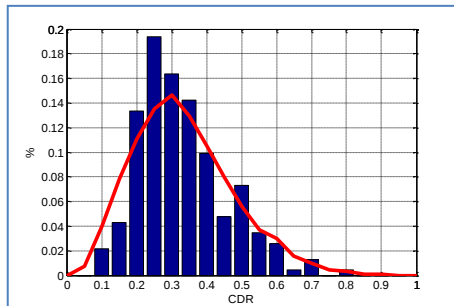
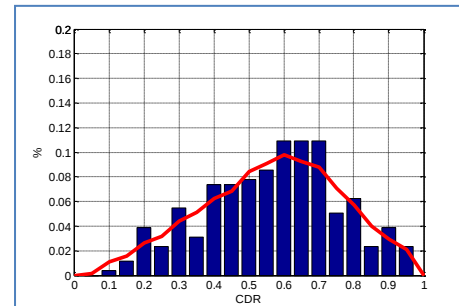


Figure 3-13 object difference and mutual area



(a) MTLn3 pEGFP ctrl



(b) MTLn3 pEGFP EGF

Figure 3-14 CDR of MTLn3 pEGFP cell line under different conditions, red lines are tend lines by moving average

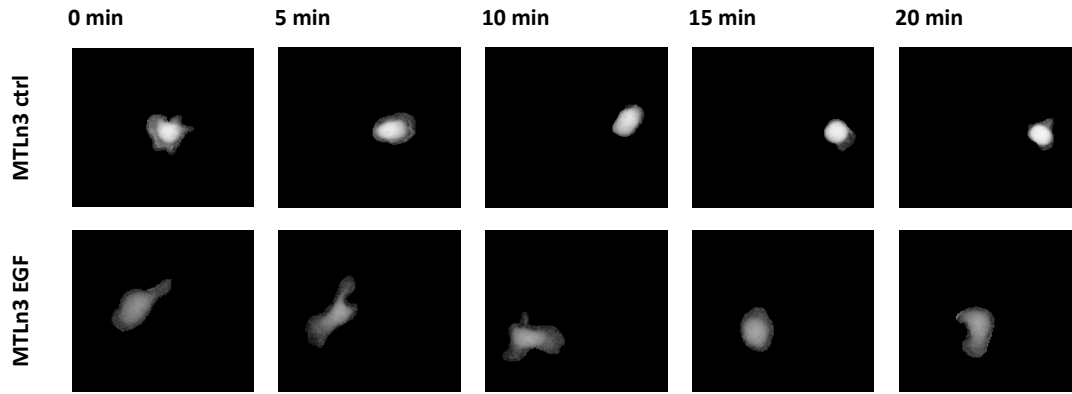


Figure 3-15 snapshots of time-lapse image sequence of MTLn3 GFP channel

Figure 3-14 shows that the CDR distribution of EGF stimulated cells significantly shifts to the right side of CDR distribution of control cells, suggesting at current temporal resolution the tracking of EGF stimulated cells becomes less efficient. It also shows that between consecutive time points EGF stimulated cells have more than 60% variation in either position or shape. Moreover, snapshots of the MTLn3 set (cf. Figure 3-15) clearly show that migration behavior of EGF stimulated cell is more aggressive and therefore difficult to model.

Subsampling Performance Estimation

To further characterize the effect of temporal resolution on tracking efficiency, a control test has been designed. The control test consists of the following steps:

1. Capture image sequence with high temporal resolution (30 second/frame)
2. Create subsampled image sequence by extracting one frame every r^{th} frame
3. Estimate CDR of subsampled image sequence (cf. Equation 3-16)
4. Perform tracking on the subsampled image sequence
5. Compare tracked path to ground truth path (cf. § 3.2.1)

The effect estimation suggests that tracking true positive slightly decreases at increasing CDR (cf. Figure 3-16). This also suggests that tracking efficiency will be affected when complexity of object motility increases. We consider KDE algorithm for tracking with different CDR. From the test result, we observe that even with a relatively high CDR, the KDE algorithm still produce a tracking result of TP = 90.36% and TN = 94.36% (cf. Figure 3-16).

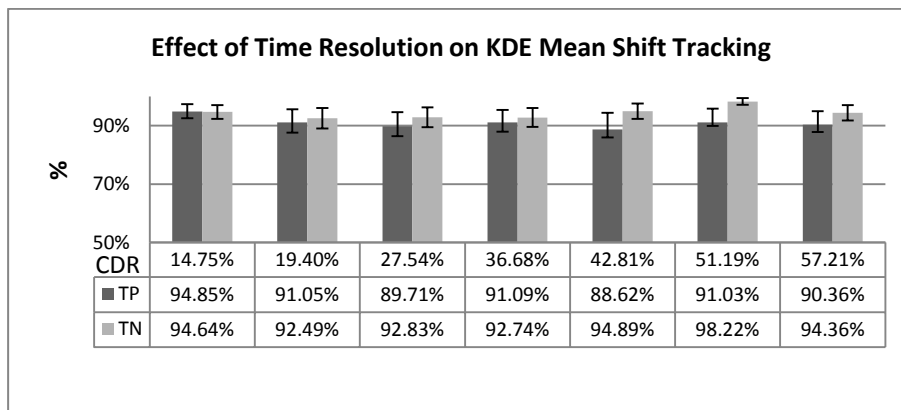


Figure 3-16 effect of time resolution on KDE mean shift tracking efficiency

3.3. Conclusions and Discussion

In this chapter, we have discussed four tracking algorithms that can be employed in the object tracking of HT/HC screens. The performance assessment of tracking algorithms shows that all tracking algorithms can produce acceptable tracking results for less motile objects such as MTLn3 pGFP cells. Our dedicated solutions, namely the KDE mean shift tracking algorithm and energy driven linear tracking can still produce a near optimal overall performance in aggressive cell behavior in an MTLn3 pGFP +EGF image set compared to other existing solutions.

Further analysis of assessment results shows that irregular and local object deformation poses difficulties for confidence measurement based tracking algorithms. The robustness of confidence measurement is a determining factor behind successful tracking. In this case, the KDE mean shift algorithm clearly outperforms the blob tracking algorithm since the kernel density estimation is more flexible and robust to small shape deformation; e.g. deformation by cell protrusions.

Compared to confidence measurement based tracking, motion model based tracking algorithms works best in object tracking when object shape information is limited, typically the matrix adhesion (MA) tracking. In the MA tracking scenario, confidence measurements between true and false candidates are less discriminative since the sizes of objects are around 5~30 pixels. However, motion model based tracking algorithm takes advantage of the presence of shape information in tracking aggressive cell migration. Moreover, the assumption of a Brownian motion model or linear motion model is not necessary true for cell migration. The major drawback behind the motion model based tracking is the assumption of a fixed motion model; increasing robustness of motion model based tracking will require flexibilization of the motion model.

From the advantages of each algorithm, we further conclude that the selection of a proper tracking algorithm should not only be based on a quality factor such as CDR. It should also be based on the availability of information on object shape and motion model. When information is limited, the choice of the tracking algorithm must be changed accordingly. Instead of considering either a confidence measurement or a motion model, a tracking solution combining both types of linkage criteria may produce more robust tracking results.

Acknowledgement

The authors would like to thank Ying Shi for her meticulous work on making the benchmark data