

Automatic Tracking of Neural Stem Cells

Tang Chunming

Information & Communication Institute,
Harbin Engineering University, China
april1971@vip.sina.com.cn

Ewert Bengtsson

Center for Image Analysis,
Uppsala University, Sweden
ewert@cb.uu.se

Abstract

In order to understand the development of stem-cells into specialized mature cells it is necessary to study the growth of cells in culture. For this purpose it is very useful to have an efficient computerized cell tracking system. In this paper a prototype system for tracking neural stem cells in a sequence of images is described. The system is automatic as far as possible but in order to get as complete and correct tracking results as possible the user can interactively verify and correct the crucial starting segmentation of the first frame and inspect the final result and correct errors if necessary. All cells are classified into inactive, active, dividing and clustered cells. Different algorithms are used to deal with the different cell categories. A special backtracking step is used to automatically correct for some common errors that appear in the initial forward tracking process.

Keywords

Tracking, stem cells, time lapse image sequences.

1. INTRODUCTION

The birth of new neurons from neuronal stem cells, a process called neurogenesis, has been seen in adult brains from both animals and humans [1]. However, little is known about the basic regulatory mechanisms of neurogenesis. In order to understand this regeneration of brain cells, cultured cells are studied. In this way, some properties of neuronal stem cells as they develop over time can be discovered. For this purpose efficient methods for tracking cells in cultures are needed.

There are a number of applications that need and use object tracking over time. We have built on methods described from some other fields, e.g. [2, 6, 7, 8], however, to get good results several special adaptations and heuristics are necessary. Estimating the motion of objects from a sequence of images consists of two major steps: (i) segmentation, i.e., separating the objects of interest from the background and from one another, and (ii) tracking, i.e., using the measurements results from segmentation to estimate the object state, which typically comprises the position and velocity [2] of the object. This paper will focus on the tracking step. We will just give a brief introductory description of the segmentation step. It should, however, be pointed out that the segmentation is quite difficult for this

application since the cells are unstained, as the stain would be harmful to the living cells and the contrast is thus quite low. Also the cell images are acquired with auto-focus which sometimes yields a poorly focused image.

Generally, the set of contiguous regions, which is the output of the segmentation algorithm, can be used for tracking. The "features" of the cells, for example, orientation, area and shape can be used to identify the different cells. However, these features are not stable [2]. Especially, the irregularly and widely varying shapes of the cells, both within an image and across a series of images, make it impossible to use masking or direct matching techniques to distinguish the cells. In our system the position of the centroids of each region in subsequent frames are considered in an initial automatic processing step and regions that can be matched are linked. Subsequent analysis steps try to link remaining unmatched regions using special heuristics. Finally the automatic tracking result can be visually checked and corrected if necessary.

2. IMAGE PROCESSING

2.1 Image acquisition

Images were captured from a computer controlled microscope attached to a cell culture system with carefully controlled environment for the cells. The time interval between images was about 15 minutes, yielding total sequences of on the average 70 frames, each illustrating the behavior of the cells under influence of a catalyzing chemical substance. The cells were unstained, and the image acquisition was completely automatic with auto-focus applied for each image frame.

2.2 Automatic segmentation

To segment the images into individual cells, we first smooth the image with a small (3x3) Gaussian filter to reduce noise. We then perform a fuzzy threshold as follows: All pixels with intensity below a lower threshold t_l are set to 0 and all pixels above a higher threshold t_k are set to 1. Between t_l and t_k image intensities are linearly rescaled to the range [0,1]. The thresholds we have chosen are for $t_l = \mu + 0.3\sigma$ i.e just above the background level. μ is the mean value and σ is the standard deviation of the background intensity. Similarly we have chosen $t_k = \mu + 4\sigma$. That is high enough

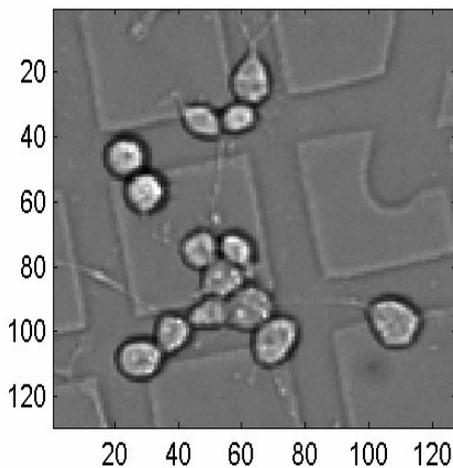
to guarantee that pixels brighter than that are really well inside the cells. Through the use of a fuzzy approach, the method becomes less sensitive to the exact values of these threshold levels, than what would have been the case if a standard crisp threshold had been used.

On the fuzzy threshold image, we apply a fuzzy gray weighted distance transform [3] to incorporate both the shape (roundness) and the intensity of the cells. This gives us a good "landscape" to segment using the watershed algorithm [4]. We use the extended h -maxima transform [5] to find suitable seed points for the watershed algorithm, where h is fixed. We require the seeds to have intensity above a threshold h in the fuzzy distance transformed image to use them for the watershed transform. In that way small and faint objects are automatically removed. After segmentation of each of the frames in the sequence of stem cell images, a series of labeled images can be obtained through application of the standard connected component labeling algorithm.

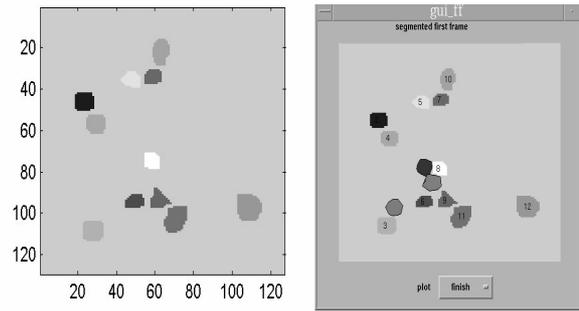
3. TRACKING ALGORITHM

3.1 Set up for the automatic tracking

Since the cell tracking is based on propagation of cell identities from the first frame throughout the entire sequence it is important to have a correct starting frame. The first image frame is shown in original gray level and segmented versions as illustrated in figure 1a and 1b, If under or over segmentation has happened it is usually obvious to human vision. The user is therefore given an opportunity of interactively correcting the segmentation as illustrated in figure 1c. The whole set of cells present in the sample will thus be tracked. At this time the operator also tells the system which cells are part of clusters.



a



b

c

Figure 1: Interactive initialization of the tracking on the first frame of the sequence.

a) Gray level image of 1st frame

b) Segmented image of 1st frame

c) Interactively corrected segmentation

3.2 The tracking algorithm for inactive cells

Most cells will only move short distances between two contiguous images [6]. Those cells are called inactive cells. For those cells a simple tracking algorithm leads to good results. This algorithm is based on the overlap between the labeled regions in the current frame and the previous one. For every cell, the algorithm first detects whether there is an overlapping region between two frames or not. If such a region is found, the algorithm records the cells ID as the same as the one in the previous image. A *tracking image* is formed from the relabeled regions in the current image using information from the previous image. This tracking image is then regarded as a new previous image. This propagates through the sequence. A table called *cells ID* which records all the assigned labels is used to guarantee that each number is used only one time in each frame.

Since the image acquisition, including focusing and segmentation is fully automatic some images will have very poor focus and the segmentation will fail resulting in serious under segmentation. This is detected through a simple test. If the number of detected cells in the current image is less than 0.15 times the number of total cells in the whole sequence, the image is skipped and the process moves to the next image. If more than three consecutive frames in a sequence have to be skipped in this way we have a serious problem and the processing is interrupted and the operator notified.

When all the overlapping regions have been matched the lists of cell ID numbers in *cells ID* of these two successive tracking images are compared. There may then be some regions in the previous frame that seems to have disappeared and some newly appearing regions that need to be handled. New appearing and disappearing *cells ID* are

saved in two 2-D matrixes, called *newappear* and *disappear* respectively.

Biologically cells do not disappear from one frame to another except when moving through the image border. Disappearing cells are thus likely to be caused by under-segmentation. In order to avoid that some cells are lost in the tracking, all disappearing cells' regions are copied from the previous frame into the tracking image except for the positions at the border of the image. When later on it is found that part of a disappearing cell's region coincides with another cell in a new tracking image, the region ID of the former is replaced by that of the latter.

3.3 How to detect and handle new cells

When moving through the sequence new cells may appear. There are three distinct reasons for this that has to be recognized and dealt with in different ways:

1. There has been a real biological cell division, creating two cells from a parent cell.
2. A cell has moved into the scene through one of the image borders
3. A cell has been split through over-segmentation

3.3.1 Detecting cell division and finding the daughter and parent cells

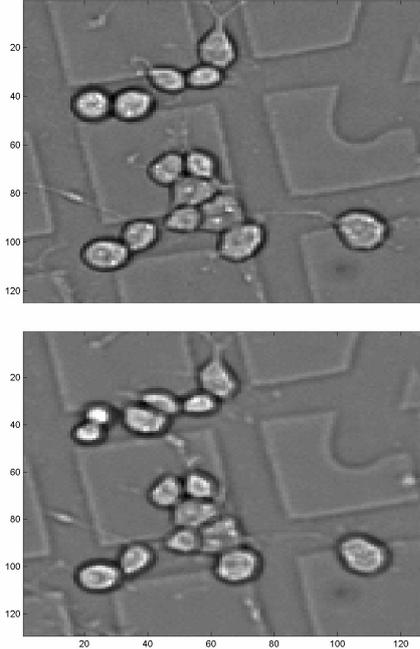


Figure 2: A scene of two contiguous gray level images with a cell division

- (a) Before division
(b) After division

It is a very important aspect of stem cell tracking to detect cell division and to save the information about the division.

For that purpose it is useful to note some biological facts that influence the cell appearance in the images. Before a cell division, the cell keeps almost stationary for several successive frames. Then it becomes very round just before the frame where the cell begins to divide because the surface tension of the cell membrane reaches maximum, as shown in Figure 2 (a). After the division, the two daughter cells are very symmetrical around the perpendicular bisector of the line linking the two centroids, as shown in Figure 2 (b). The area of the parent cell is almost equal to the sum of the two daughters' areas. One daughter cell keeps its parent cell's original position while the other one is pushed away some distance.

We try to use these features in an analysis of each new cell region that appear during the tracking to see if it can be a cell division. First of all, we use the cell position to detect whether the new j_{th} cell in the 2-D matrix *newappear* in 3.2 comes from the border or not. If not, we consider whether the new one comes from a division. All distances between the j_{th} cell and the others are computed (we define distance as the length of the line linking the two cells' centroids). If the distance to the i_{th} cell is less than D_{fast} , the i_{th} cell becomes a candidate in C_i , as the other cell in the pair after division.

To verify if the candidate truly represents a division, a number of conditions that are heuristic parametric versions of the observations in the previous paragraph are applied. We accept that the new j_{th} and i_{th} cell in C_i is a division pair in the k_{th} tracking image if the following conditions are satisfied. For each candidate i in C_i :

- C1. Distance (1)

$$abs(D_{i,j} - 0.5(\max(L_{i,max}, L_{j,max}) + \max(L_{i,min}, L_{j,min}))) < 1$$

$$D_{i,j} : \text{Distance between cell } i \text{ and } j$$

$$L_{i,max}, L_{i,min} : \text{Major and Minor Axis Length of } i$$
- C2. Area (2)

$$abs(S_{k-1,i} - (S_{k,i} + S_{k,j})) / S_{k-1,i} < 0.3$$

$$\& \quad abs(S_{k,i} - S_{k,j}) / \max(S_{k,i}, S_{k,j}) < 0.04$$
- C3. Bounding Box (3)

$$abs(\text{BoundingBox}(i) \bullet x_width - \text{BoundingBox}(j) \bullet x_width) \leq 1$$

$$\& \quad abs(\text{BoundingBox}(i) \bullet y_width - \text{BoundingBox}(j) \bullet y_width) \leq 1$$
- C4. Convex Area (4)

$$abs(\text{ConvexArea}(i) - \text{ConvexArea}(j)) \leq 2$$
- C5. Equiv Diameter (5)

$$abs(\text{EquivDiameter}(i) - \text{EquivDiameter}(j)) < 0.5$$
- C6. Solidity (6)

$$abs(\text{Solidity}(i) - \text{Solidity}(j)) < 0.09$$

3.3.2 Save the information on cell division

As every cell has its unique cell ID, we must save the information about the division so that we can know the behavior and regeneration of stem cells in the whole sequence. For this purpose we create a data table with fields: cell route and frame number, e.g.: *Table (2): cell route: [11 17], division frame: [15]*

The example means that a division happens in the 15th frame which produces a new cell 17 whose parent cell is cell 11 in the 14th frame. This is the second division in this sequence, thus saved in *Table (2)*.

3.4 The tracking algorithm for active cells

If the new appearing j_{th} cell neither is coming from the border nor is meeting the conditions of a division cell, then we check whether it is an *active cell* which could not be detected by overlap. Active cells are defined as cells that move rapidly, more than D_{slow} pixels per frame. Biologically they are of special interest. All the disappearing cells in the current frame registered in the 2-D *disappear* matrix $C_{d(i)}$ described in 3.2 become candidates in order to match the active cell. To reach the best matching, features are computed for each candidate. The following formula is used to find the best candidate [7]:

$$C_{min} = \underset{Cd(i)}{\operatorname{argmin}} f(C_{d(i)}) = \underset{Cd(i)}{\operatorname{argmin}} \{ \alpha D_{j,Cd(i)} + \beta A_{j,Cd(i)} + \gamma P_{j,Cd(i)} \} \quad (7)$$

In formula (7), $D_{j,Cd(i)}$ is the distance between the i_{th} candidate and the new appearing j_{th} cell. $A_{j,Cd(i)}$ is the difference in area between the i_{th} candidate and the new appearing j_{th} . $P_{j,Cd(i)}$ is the difference in perimeter between the i_{th} candidate and the new appearing j_{th} . The α , β and γ are weights. Finding the minimum $f(C_{d(i)})$ means finding the cell in the *disappear* matrix that best matches this new appearing j_{th} cell. Among the three factors, the distance is the most important. The other two weights depend on the segmentation algorithm. We have used $\alpha=0.6$, $\beta=0.2$, $\gamma=0.2$. The region label for the j_{th} new cell ID is changed to the best matching candidate's ID.

3.5 Estimation of cell speed parameters

There are two important parameters in the tracking algorithm relating to cell speed: D_{fast} and D_{slow} . These parameters are based on a model of the cells behavior in terms of Brownian motion [8]. In a homogeneous 2-D system, the probability distribution C that a cell with a diffusion coefficient D suffers a displacement r in a time period τ is:

$$C(r, \tau, D) = \left(\frac{1}{4\pi D \tau} \right) \exp\left(-\frac{r^2}{4D\tau} \right) \quad (8)$$

Here, $D = \frac{R}{3\pi\eta R_H}$. In our experiment τ , the sampling time

is 15 minutes, η , the viscosity coefficient for 10% fetal bovine serum, is $(2\sim 5) \times 10^{-3}$ Ns/m², R is 8.3144 Jmol⁻¹, and

L is 6.0221×10^{23} . According to formula (8), we can fit a curve to the sampling points which stand for the cells motion between two contiguous frames, as shown in Figure 3. From Figure 3, we can see that the distance moved by the majority cells is no more than 10 pixels. We thus use this value for D_{slow} . And the distance moved for active cells is no more than 18 pixels which is our value for D_{fast} .

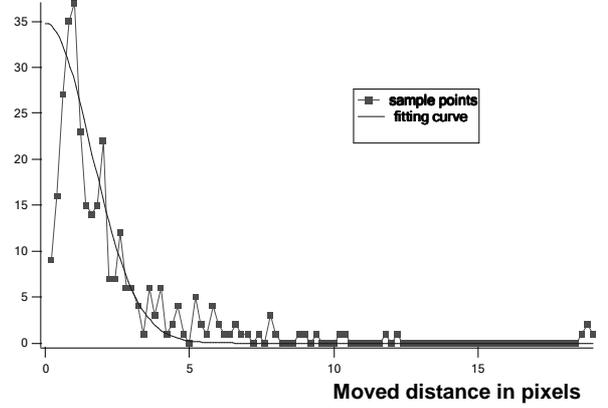


Figure 3: Fitting a Brownian motion model to the cell motion data

3.6 Backtracking

After performing the cell tracking in a forward matter as described so far, some errors may have appeared that can be detected and corrected through some backtracking through the sequence.

3.6.1 Exchanging cell ID in two successive tracking frames

When some cells disappear due to under-segmentation and some move a lot because they are being active, the previous algorithms may not work properly. The most apparent mistake is that two cells exchange their ID in two successive tracking frames. A special processing step will detect and correct this error:

Build a subset of all the cells in the $k_{th} + 1$, tracking image that are more than D_{slow} pixels apart by scanning all cells between two contiguous tracking images after the process in 3.4. Study the orientation of the distance vector between cell pairs in this subset to see if any two has similar magnitude and opposite directions. Those are likely to have been exchanged by mistake and this can be corrected by swapping the ID numbers. The following figure illustrates this for region numbers 5 and 10.



Figure 4: Model of exchanging cell pair

3.6.2 Feedback about the static cells

Some cells will not appear to move at all between several successive frames in the sequence. Such cells are called static cells. The reason may be either biological or a processing artifact.

There are some cells that are static because of the replacement described in 3.2. The reasons are over-segmentation or some cells moving in from the border then out again after several frames. We set up a 2-D matrix called *static_v* which is formed by frame number and *cells-ID*. It records the ID numbers of all cells that keep static positions between two successive frames. For each frame after the third one in the sequence we compute for each cell C_{st} in *static_v*, the number called t_{static} of successive frames that the cell stays static.

If $3 \leq t_{static} \leq 4$, we find the first frame where this cell became static. Then we check whether there are some new cells C_{new} appearing in this frame (including the cells moving in from the border but not including the cells from division). If the distance $|C_{new} - C_{st}| < D_{fast}$, the region of the static cell is updated to zero while the region of the new cell is updated to the ID of this static one in all frames where this static cell exists. After this, the corresponding values in *static_v* are set to zero. If no new cell appeared in the first frame, the static cell is kept for another few frames.

If $t_{static} \geq 5$, and there are some static cells that were not marked as belonging to clusters in the initial process described in section 3.1, these static cells are deleted for all these frames. After that, the corresponding values in *static_v* are set zero. These static cells are regarded as coming from over-segmentation.

4. RESULTS AND CONCLUSION

Figure 5 and 6 illustrate two successful applications of the described procedure: creating a complete trace of a set of neural stem cells over 71 frames. As shown in figure 5, we can observe that cell 13 divides into two cells after some time, indicated by the two red stars and the black stars which represent the two traces after cell 13 divides. We also notice that cell 16 moves out through the right hand border of the image but then returns some frames later.

Figure 6 shows a slightly different representation of the results of another experiment. Here the traces marked with a triangle are daughter cells after split. These traces do not include cells that remain clustered. The plot shows some errors due to under-segmentation causing unrealistically large distances between points, such as, 1 to 2 and 2 to 3.

The described system was a pilot study intended to demonstrate how an effective tool for automatic cell image tracking could be designed, including a possibility for the user to interactively supply crucial information and correct errors. This objective has been fulfilled. One of the merits of the system is that it tracks all cells present in the cell culture, not just a few selected ones.

The system still has some limitations. The judgment about when to split cells could be made more accurate through the use of more advanced shape analysis. There is also a possibility of using feed-back from the tracking stage to the segmentation stage, i.e. to look more carefully for cells in regions where the tracking algorithm suggests there should be a cell but the initial segmentation did not find one.

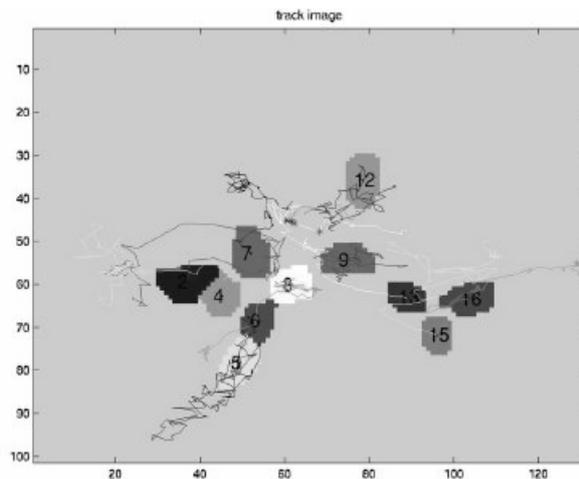


Figure 5: Trace of cells in a whole sequence of 70 frames

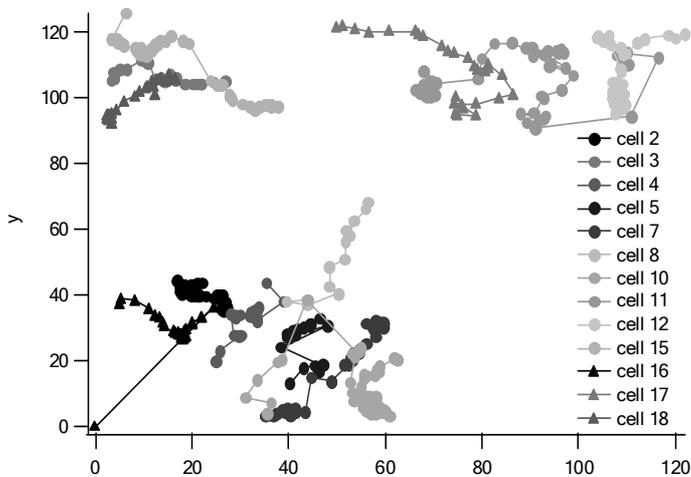


Figure 6: Trace of cells in another sequence of 71 frames

The system including its user interface has been implemented in Matlab 6.5 which has been convenient for the experimental development. A new stand-alone implementation is currently being developed to increase the processing speed and decrease memory requirements when very large data sets need to be processed. The new version will also include some of the improvements this prototype has shown possible and desirable

5. ACKNOWLEDGMENT

The authors gratefully acknowledge Joakim Lindblad and Carolina Wählby of the Center for Image Analysis for developing the segmentation algorithms and Johan Degerman of Chalmers University for recording the stem cell image sequences.

6. REFERENCES

1. P. Eriksson et al., "Neurogenesis in the adult human hippocampus", *Nat. Med.*, 4, pp. 1313-1317, 1997
2. T. Kirubarajan and Y. Bar-Shalom, "Combined segmentation and tracking of overlapping objects with feedback," *Multi-Object tracking, 2001 IEEE Workshop on*, pp. 77-84, Vancouver, July 2001
3. P. K. Saha and F. Wehrli and B. R. Gomberg, "Fuzzy distance transform: theory, algorithms, and applications," *Computer Vision and Image understanding*, pp. 171-190, vol. 86, no.3, 2002
4. L. Vincent and P. Soille, "An efficient algorithm based on immersion simulations," *IEEE Trans. on Pattern Anal. and Machine Intelligence*, pp. 583-597, vol. 13, no.6, 1991
5. P. Soille, *Morphological Image Analysis: Principles and Applications*, Springer-Verlag, October 1999
6. T. Kirubarajan and Y. Bar-Shalom and K. R. Pattipati, "Multiassignment for tracking a large number of overlapping objects with feedback," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 37, no.1, Jan. 2001
7. Xia Li, Mingming Hao, David W. Piston, Benoit M. Dawant, "Automatic tracking of proteins in sequences of fluorescence images," *Proceedings of SPIE---Volume 5370, Medical Imaging 2004: Image Processing*, pp. 1364-1371, May 2004
8. Stephane Bonneau, Laurent Cohen, Maxime Dahan, "A multiple target approach for single quantum dot tracking," *IEEE International Symposium on Biomedical Imaging, ISBI*, April 2004