Cellular tracking in time-lapse image data (August 2013)

Vincent Andrearczyk, M.Eng, Dublin City University

Abstract— **Automatically tracking cells in large time lapse datasets is necessary for analyzing cell motion and behavior in order to develop new diagnostics and therapeutics. Algorithms combining a frame-by-frame segmentation and a model-based method have been developed [1][2] to track the cells across a sequence of images. The proposed method is based on this approach, combining a frame-by-frame method (cell detector and arbitrator) and a model-based method using the Active Shape Model (ASM). As far as the obtained results are concerned, we developed a robust cell tracking algorithm that can track the cells across a sequence of cell population images.**

***Index Terms*— image processing, cell tracking, Active Shape model**

I. Introduction

Tracking of cell populations provides important and relevant information on the cell behaviors that can be used for a significant amount of applications in genomics, proteomics, stem cell biology and tissue engineering. Cell analyses are used to develop new diagnostics and therapeutics by analyzing their behavior and reaction in different environments. The data load of cell image sequences available for analysis is too important to be analyzed manually. Automated cell tracking algorithms are therefore essential in order to substitute the manual marking of cells across a sequence of images. Several issues occur in the cell tracking process, such as the mitosis (division of a cell into two new cells), the apoptosis (cell death), the cells entering and leaving the field of view of the camera, the change of cell shapes, the increasing density during expansion and the overlapping cells.

The main traditional methods developed in prior works are classified in two principal processes: tracking by detection and tracking by model-evolution [1][2]. The tracking by detection method operates a frame-by-frame segmentation followed by an inter-frame matching whereas the tracking by model evolution utilizes a mathematical model that fits the cells and that evolves overtime to follow the changes of location and shape. Both techniques entail advantage. The first approach enables the detection of the mitosis and of the cells entering and leaving the field of view whereas the second can deal with dull edges. K. Li et al. [1] present a technique that combines these advantages. Basic techniques such as thresholding or Hough transform [8] have been used for segmentation as well as more advanced methods as for instance multifractal [4][6] and pattern recognition [7]. A neural algorithm was applied in [5] to deal with noisy cells. In addition, a genetic algorithm in [6] improves the separation quality of the segmentation.

We propose an algorithm that combines the tracking by detection and tracking by model-evolution approaches. In our technique, the tracking by model-evolution uses the ASM to match the cells over the sequence. Moreover, an arbitrator and a cell detector allow us to deal with the cells entering and leaving the field of view and to manage the lost tracks.

The solution proposed in this paper has revealed good results with regards to the efficiency and robustness of the cell detection and tracking. 89% of the cells are accurately detected by the cell detector and 92.1% of the detected cells are tracked between two consecutive frames. This automated analysis enhances the cell population movement that can be used in order to analyze the behavior of the cells, their proliferation or their health.

II. Existing techniques for cell tracking

Existing cell tracking techniques are essentially classified into two approaches: tracking-by-detection and tracking by model-evolution, each entailing specific advantages and disadvantages.

The tracking-by-detection is processed in two independent stages: a segmentation is processed on all the images using thresholding, gradient, Hough transform [8], multifractal [4] [6] or pattern recognition [7] in the first stage.

In the second stage, an inter-frame data association is processed in order to match the cells frame after frame. The segmentation in each frame can be independent from the other frames, making it possible to deal with mitosis and cells entering and leaving the field of view.

The tracking by model-evolution approach consists of creating a mathematical appearance or shape model and of optimizing its parameters in order to match the cells across the sequence. Typical models used for cell tracking are Active Contour Models (ACM or “snakes) [9], ASM or Mean-shift [10]. Unlike the tracking by detection method, this kind of approach does not require an inter-frame object pairing. Thus, it allows the cell tracking to deal with dull edges and overlapping issues.

Several researches have been conducted in order to combine the advantages of these two methods. K. Li et al. [1] present such an approach including modules designed in two levels: a cell detector and a track arbitrator. However, this solution also shows limitations regarding the spatiotemporal information that is lost by processing in a frame-by-frame manner.

Neural classifiers have also been applied to deal with noisy images, whose results have revealed an efficient learning of classification in the case of blurred and non-sharp cell shapes [5]. In [6], an optimization of the segmentation is developed using a genetic algorithm. More information on the prior work may be found in [11].

III. Cell tracking algorithm description

Our algorithm is composed of two main parts: the training program and the tracking program. While the first part is run once in order to save the possible shapes of the cells (ASM space), the tracking program is run on a new sequence of an image, performing the cell detection and tracking. The ASM search is the principal element of our approach and is derived from the ASM shared code in [3]. The second element that supplements the ASM search is an inter-frame matching of the cell contour.

A. Training process

The training process is required in order to save the ASM space which represents the possible variations of the cell shapes as well as to create an appearance model that defines constraints for the ASM fitting optimization. To this end, a training set composed of 80 cell contour shapes that have been manually marked is created. The main code of the training process is available at [12].

At the first step, the possible shapes and the mean shape are learned and saved. To this end, the rotations and the translations are removed from the cell contours (training set). The Principal Component Analysis (PCA) is consequently used to determine the most important and uncorrelated eigenvectors and eigenvalues of the training set. The PCA uses the Single Value Decomposition to find the eigenvalues and eigenvectors. At last, the shape space is saved. It is composed of the eigenvectors and the eigenvalues, both describing the possible shapes of the cells, as well as of the mean training shape.

Secondly, an appearance model is created and saved in order to optimize the ASM fitting. For each contour, the image is therefore sampled on lines perpendicular to it. Fig. 1 shows the mean intensity profile of these lines for each contour. We can observe the higher intensity at the contour point, which lies on the 9th point of the x-Axis for each contour. This constant particularity of its perpendicular lines will be used in order to find the best fit for the ASM search. This information is saved in two ways in order to have two possibilities for the ASM fitting optimization. The first method uses the Mahalanobis distance. The covariance matrix of the derivative of the intensity profile is thus saved to enable this measure. The second method uses the PCA on the intensity profile. Therefore, the eigenvectors, eigenvalues and mean of the intensity profile are saved. This process is applied for different scales of the image in order to refine the optimization of the ASM fitting. The intensity profile are calculated on the original image and on rescaled images. The eigenvectors, eigenvalues, mean and covariance matrix are subsequently saved for each scale.

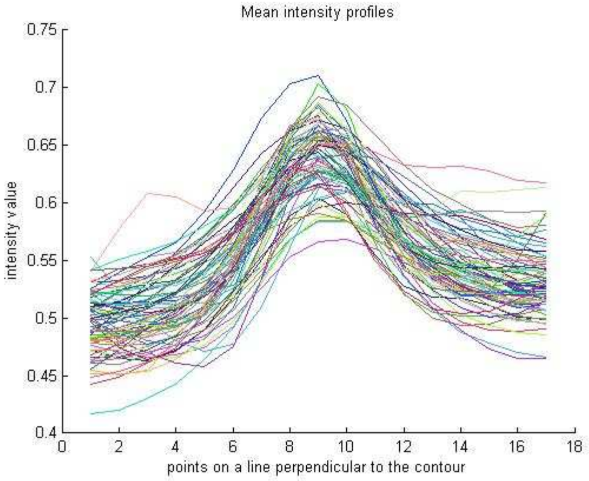

Fig. 1. Mean intensity pixel profile of the lines perpendicular to the contours and passing through the contour points. Each of the 80 curves corresponds to a contour that has been manually marked

B. Cell tracking process

Once the training process has been run for the appropriate type of cell images, the cell tracking system can be processed on a new sequence of cell images. Our tracking algorithm is essentially composed of a cell detector, a cell tracker and an arbitrator. The main code of the cell tracking algorithm is available at [13].

The cell detector locates the cell contours on an image of the sequence using the ASM fitting. The initialization of the ASM is based on simple binary operations and on a threshold applied on the input image, which calculates the potential centers of the cells. An example of this initialization may be found in [14]. The initial shape of the ASM is the mean shape of the training data that is loaded at the beginning of the program. As mentioned in the previous part, the ASM fitting can be optimized in two ways: by minimizing the Mahalanobis distance or by calculating the PCA parameters. The search of the best ASM fitting is applied on different scales of the image.

The cell detector is applied to all the cells of the first image of the sequence as well as on the cells of the following images that have not been tracked (see figure 1).

In the first case, the cell detector detects all the cells of the first image in order to start the tracking.

In the case of the latter, this is used in order to detect cells that have not been tracked from a previous image. Detecting cells that have not been followed from a previous frame present two advantages. First, we seek to match more cells using an inter-frame cell matching based on their positions and shapes. Cells whose tracks have been lost from the previous image are therefore followed again. Second, the detected cells are used in addition to the previously tracked cells for the cell tracker initialization on the next frame. Cells entering the field of view or cells which were not previously detected and tracked can therefore be managed by the tracking system.

The cell tracker applies the same ASM search as the cell detector, yet its shape and position initializations are based on the positions and shapes of the cells detected or tracked on the previous image. It is run on all the frames of the sequence except for the first frame since no previous initialization is available (Fig. 2). A cell is considered followed or "tracked" between two consecutive frames if the ASM search initialized by the previous cell contour fits the cell on the present frame again. We assumed and verified that the changes of cell shapes between two consecutive frames were slight enough to consider that the ASM search would fit the contour of the same cell as on the previous image.

The track arbitrator block specifies whether the ASM results (cell contours) are consistent or not. The contours which are considered as not consistent by the arbitrator are removed from the detected or tracked cell contours set. In order to do so, different modules analyze the cell contours and the image.

First, those cell contours fitted by the ASM search, which partly lie outside the field of view, are removed.

Based on the gradient values of the image, and on binary operations, a mask is created in the second step, aiming to describe the areas of the image that cannot contain a cell. Those areas indeed possess a low pixel variation and are therefore easily extractable. In the arbitrator, the cell contours which have a part in this mask area are removed.

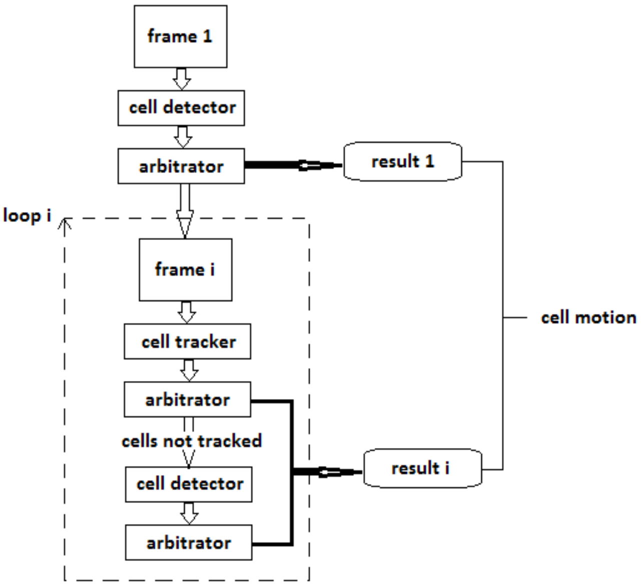

Fig. 2.Structure of the cell tracking algorithm resulting in the cell motion information

The third module tests the centers of the cells used for the initialization of the cell detector. Cell contours containing zero or more than one of these centers are removed.

In the last module, a cell contour that contains the same center as another contour is deleted as well. The arbitrator is applied after the cell detector and the cell tracker (Fig. 2).

In the end, another module that is not represented on Fig. 2 for the purpose of a clearer representation is used for newly matching the cells that have neither been tracked nor matched on two previous frames. Some cell contours from the frame (i-2) may not be on the frame (i-1). This module simply compares these contours to the cell contours on the frame (i). The cell tracking is therefore more persistent, as the track of a cell that is lost on a frame can be recovered later in the sequence.

For each image, the centers of the cells are saved as well as numbers specifying which cells have been successfully tracked from the previous image. At the end of the analysis of the sequence, all the information on the motion of the cells that have been detected by the system has been gathered. For a clear result, we only show the motion of the cells that have been tracked across the whole sequence.

IV. Cell tracking results

Our program was run on four sequences of 100 frames each. The training ASM space was saved once and used for the four sequences as the cell images in large time-lapse dataset were acquired in a similar manner and context. As explained previously, two optimizations are available for the ASM fitting. The PCA method has shown much better results than the Mahalanobis distance minimization. The following results are therefore obtained by using this method. Fig. 3 shows the ASM search of the cell detector on a single cell.

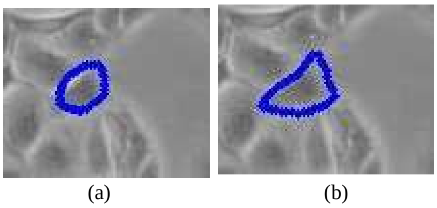
(a) (b)

Fig. 3. Example of ASM search of the cell detector (a) Initial ASM position and shape. The shape is the mean shape of the training data, (b) ASM fitted to the cell contour

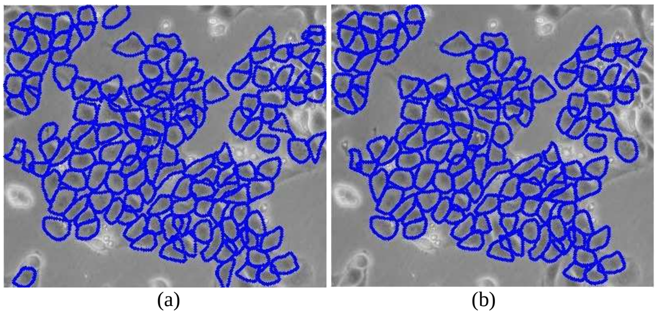
(a) (b)

Fig. 4.Example of ASM search showing the detection of all the cells on an image (a) before the Arbitrator (c) After the Arbitrator

The process of the arbitrator module is shown on Fig. 4. We see that the cell contours that are considered not consistent as explained in the previous part are removed by the arbitrator such as those contours lie outside the field of view.

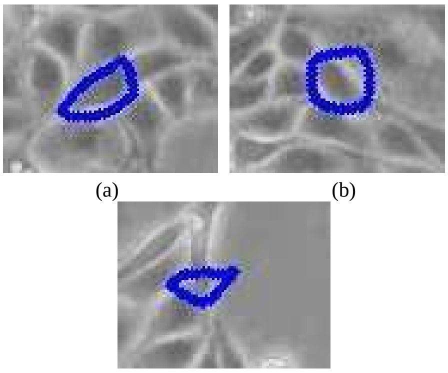

Fig. 5. Example of ASM search of the cell detector (a) good detection, (b) imprecise detection, (c) wrong detection

The cell detector has been tested on twelve cell images (three from each sequence). We have manually determined the number of cells on each image and analyzed the ASM result (cell detector).

An average of 137.7 cells has been manually found on each image. After the arbitrator correction, 94.2% of these cells have been matched by the cell detector, 2.72% of these cells were wrong, 8.52% of the actual cells were therefore missed on average and 4% were “not accurate”. Eventually, 88.6% of the cells were precisely detected on average on each frame. A shape is considered wrong if it does not fit a cell contour or if it fits two cell contours. A shape is termed “not accurate” if we clearly see that it does not precisely fit a cell contour (see Fig. 5).

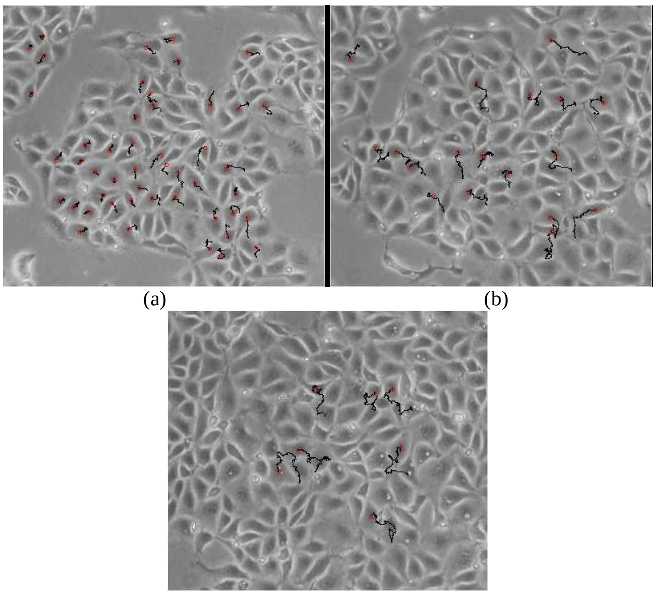

Fig. 6. Result of the cell motion over a sequence of (a) 20 frames, (b) 50 frames, (c) 100 frames

Tested on twenty cell images (5 from each sequence), the percentage of cells tracked by the cell tracker - after correction by the arbitrator - was 89.3% without the module that recovers the track of the lost cells. It was 90.42% with this module dealing with the lost cells from one frame to the other and 92.1% when using the module, which deals with the lost tracks from frame (i-2) to frame (i). The tracked cell percentage only represents those cells detected by the ASM search; it does not represent the real percentage of cells tracked on the image.

Eventually, Fig. 6 shows the track (black marks) of those cells that have been followed from the first image until the end of the sequences composed of 20, 50 and 100 frames. However, more material is saved for analysis as all the information about the tracked cells is saved.
The final result therefore presents a table of vectors that contain all the centers of the cells detected by the system for each frame as well as a vectors specifying which cells have been followed on two consecutive frames.

Further results and tests may be found in [14]. Among other things, the ASM optimization is tested for different number of scales.

V. Analysis

A. *Robustness and precision of the tracking*

As outlined in the previous part, we developed a robust cell tracking algorithm. The detection and tracking of the cells is more robust for the cells that present a simple circle shape entailing a medium or big size. As we can see on Fig. 6, only a few cells are tracked from the first image until the end of the sequence. This is due to the fact that some cells leave the field of view, others may die (apoptosis) or separate (mitosis) and cause a lost track. Some cells may simply not be fitted by the ASM because of their shapes, causing further lost tracks. Those explain the few track traces on Fig. 6 (48 traces over 20 frames, 18 traces over 50 frames and 7 traces over 100 frames). However, more information can be extracted from the analysis. The result of the algorithm analysis indeed contains all the cell centers that have been detected on all the frames of the sequence with a detection percentage of 90.14%. On top of that, the cells that are followed on every two consecutive frames are further specified. The information of nearly all cell movements is therefore saved and available for analysis.

The ASM search shows several difficulties on certain types of cell shapes.

First, while the cell detection and tracking are indeed more accurate for cells of medium and large size, small cells are harder to fit since the ASM search tends to grow and to fit two cells instead of a single one. The reason for this is that most of the shapes are bigger and therefore the mean shape is bigger, causing the ASM search to match two small cells that together present a shape similar to a single cell shape.

Secondly, the detection of those cells that are not surrounded by other cells is also less robust and accurate. Indeed, the ASM search fits to the intersection between two cells more easily than to the intersection between a cell and the background of the image. This is explained by the optimization of the ASM search that uses the mean intensity profile of the lines perpendicular to the contours. Most of the contours being surrounded by other cells, the ASM searches pixel values that correspond to another cell around the contour. These values are not found in the case of cells not surrounded by other cells and the ASM fitting is therefore less accurate.

However, in both of these cases, the cell detection is wrong and is therefore removed by the arbitrator as explained in part III. Hence, these fitting errors cause a non-detection of the cells instead of an imprecise or wrong detection.

Furthermore, since we cannot determine the exact position and shape of those cells that are not entirely on the image, they are also removed from the cell tracking in the arbitrator process in order to optimize the precision of the cell tracking at the cost of fewer cell detections.

The final results have shown a precise detection of 90.14% of the cells by the cell detector and have further revealed that 92.1% of the detected cells are tracked between two consecutive frames by the cell tracker. It should be noted that the percentage of cells that are not detected or not tracked represents mostly the same cells on different images, which signifies that most of the cells are tracked across the sequence for a long duration and enhance the cell population motion and behaviour.

B. *Further development of the system*

With regards to the mitosis detection, the segmentation is not precise enough to implement a robust system. Indeed, we need exact cell shape detection in order to detect the cell changes that occur with the mitosis and especially a good detection of small cell shapes is required as daughter cells are smaller than others. Explanation of the mitosis phenomena may be found in [11].

We have also not implemented a prediction filter based on the Optical Flow as the cell motion was minimal and unpredictable between two images. Explanations of the Optical Flow may be found in [11]. The tracking issues were not due to the fact that a cell was not followed from an image to the following, but to the fact that the ASM search could not fit the cell shape. In this context, it was not helpful to use a prediction filter.

VI. Conclusion

We developed an automated algorithm capable of tracking hundreds of cells in large time-lapse image sequences of an increasing cell population. The system incorporated an ASM based process as well as a frame-by-frame cell detection and matching. The system achieved a detection of 90.14% of the actual cells on the frame and a tracking of 92.1% of the detected cells from one frame to the other. The results obtained provide useful information on the cell movement and enable studying the cell population behavior. Further improvement on robustness, processing speed and mitosis detection may be developed based on this cell tracking algorithm.

References

[1] K. Li, E. D. Miller, L. E. Weiss, P. G. Campbell, T. Kanade, “Online Tracking of Migrating and Proliferating Cells Imaged with Phase-Contrast Microscopy”, *Proc. Conf. Computer Vision Pattern Recognition Workshop*, CVPRW, 2006, pp. 65-72.

[2] K. Li, M. Chen, T. Kanade “Cell population tracking and lineage construction with spatiotemporal context”, *10th Int. Conf. Medical Image Computing Computer Assisted Intervention*, 2007, pp. 295-302

[3] D.-J. Kroon (2010) , *Shared code “Active Shape Model (ASM) and Active Appearance Model (AAM)”* [online], available: http://www.mathworks.com/matlabcentral/fileexchange/26706-active-shape-model-asm-and-active-appearance-model-aam

[4] N. Lassouaoui1, L. Hamami and A. Zerguerras, “Segmentation and classification of biological cell images by a multifractal approach”, *Int. J. Intelligence Systems*, vol. 18, pp.657-678, June 2003

[5] T.W. Nattkemper, H. J. Ritter and W. Schubert, “A Neural Classifier Enabling High-Throughput Topological Analysis of Lymphocytes in Tissue Sections”, *Transaction Information Technology Biomedicine*, T-ITB, 2001, pp. 138-149

[6] N. Lassouaoui, L. Hamami , “Genetic algorithms and multifractal segmentation of cervical cell images”, *7th Int. Symp. Signal Processing Its Applications*, 2003, pp. 1-4

[7] P. Spyridonos, D. Glotsos, D. Cavouras, P. Ravazoula, V. Zolota and G. Nikiforidis, “Pattern recognition based segmentation method of cell nuclei in tissue section analysis”, *14th Int. Conf. Digital Signal Processing*, DSP, 2002, pp. 1121-1124

[8] P. P. Guan and H. Yan, “Blood cell image segmentation based on the Hough transform and fuzzy curve tracing”, *Proc. Int. Conf. Machine Learning Cybernetics*, ICMLC, 2011, pp.1696-1701

[9] N.Ray and S. Acton “Active Contours for Cell Tracking”, 5 th Proc. of the IEEE Southwest Symposium Image Analysis Interpretation, SSIAI, 2002, pp. 274-280

[10] O. Debeir, P. V. Ham, R. Kiss and C. Decaestecker, “Tracking of migrating cells under phase-contrast video microscopy with combined mean-shift processes”, Transaction Medical Imaging, T-MI, 2005, pp. 697-711

[11] V. Andrearczyk, “Cellular tracking in time lapse image data Literature Survey”, Appendix A, April 2013

[12] V. Andrearczyk, “Main code of the training algorithm”, Appendix B, August 2013

[13] V. Andrearczyk, “Main code of the cell tracking algorithm”, Appendix C, August 2013

[14] V. Andrearczyk “Further tests and results”, Appendix D, August 2013

[15] Paul F Whelan (2011) “VSG Image Processing and Analysis (VSG IPA) Toolbox”, Technical Report, Centre for Image Processing & Analysis, Dublin City University.

[16] Paul F Whelan (2011) “VSG Image Processing and Analysis (VSG IPA) Matlab Toolbox - Software””, http://www.cipa.dcu.ie/code.html, Centre for Image Processing & Analysis, Dublin City University.

[17] Paul F. Whelan and D. Molloy (2000), "Machine Vision Algorithms in Java: Techniques and Implementation", Springer (London), 298 Pages. ISBN 1-85233-218-2.

[18] Paul F Whelan (2011), "EE544: Computer Vision Course Notes”, Centre for Image Processing & Analysis, Dublin City University.

[19] Paul F Whelan (2011), "EE425/EE453: Image Processing and Analysis Course Notes”, Centre for Image Processing & Analysis, Dublin City University.