33rd Annual International Conference of the IEEE EMBS
Boston, Massachusetts USA, August 30 - September 3, 2011

New robust algorithm for tracking cells in videos of drosophila morphogenesis based on finding an ideal path in segmented spatio-temporal cellular structures

Yohanns Bellaïche, Floris Bosveld, François Graner, Karol Mikula, Mariana Remešíková, Michal Smíšek

Abstract— **In this paper, we present a novel algorithm for tracking cells in time lapse confocal microscopy movie of a Drosophila epithelial tissue during pupal morphogenesis. We consider a 2D + time video as a 3D static image, where frames are stacked atop each other, and using a spatio-temporal segmentation algorithm we obtain information about spatio-temporal 3D tubes representing evolutions of cells. The main idea for tracking is the usage of two distance functions - first one from the cells in the initial frame and second one from segmented boundaries. We track the cells backwards in time. The first distance function attracts the subsequently constructed cell trajectories to the cells in the initial frame and the second one forces them to be close to centerlines of the segmented tubular structures. This makes our tracking algorithm robust against noise and missing spatio-temporal boundaries. This approach can be generalized to a 3D + time video analysis, where spatio-temporal tubes are 4D objects.**

I. INTRODUCTION

Cell tracking in a developing organism means extracting spatio-temporal trajectories of cells and detecting moments of cell divisions. It is one of the most interesting topics in the modern biology - a reliable backward tracking method could answer some of the fundamental questions of developmental biology: global and local movement of cells, origin and formation of tissues and organs, cell division rate and localization, etc.

In this paper, we present a new method for tracking cells in 2D + time image sequences. We consider a time sequence of 2D images as a 3D image, where separate frames are stacked atop each other. We identify cell evolutions as a set of spatio-temporal tubes. We achieve this via spatio-temporal 3D segmentation. Having these tubes segmented, tracking means, from a given point in tube interior, to find a trajectory - within this tube - to the cell identifier in the first video frame. In later sections, we will refer to these first-frame cells as to the "root cells". Finding a correct trajectory is achieved via computation and use of two constrained distance functions. The distance function from root cells forces trajectory to approach a root cell. The distance from segmented boundaries keeps this trajectory centered.

Y. Bellaïche, F. Bosveld and F. Graner are with UMR 3215, U934, Institut CURIE, 26 rue d'Ulm, 75248 Paris Cedex 05, France {yohanns.bellaiche, floris.bosveld, francois.graner}@curie.fr

K. Mikula, M. Remešíková and M. Smíšek are with Faculty of Civil Engineering, Slovak Technical University, Radlinskeho 11, 81368 Bratislava, Slovakia mikula@math.sk, remesikova@math.sk, michal.smisek@gmail.com

We test the algorithm on a video of the mono-layered epithelium of the *Drosophila* pupa, which undergoes extensive proliferation and morphogenesis to form the *Drosophila* adult. Upon expression of E-Cadherin-GFP, which localises the adherent junctions, its development can be followed by confocal time-lapse microscopy [1]. The video was acquired with Nikon Ti spinning disk microscopes equipped with a HQ2 Ropper Camera in routine conditions for these kinds of movies. It consists of 199 frames, has resolution 569 x 500 pixels and intensity ranges from 0 to 255, the cells often divide but never merge and they also move to the right. In fig. 1 one can see visualization of the image data.

II. ALGORITHM STEPS

Our algorithm consists of these consistent, but independent steps:

A. Cell identification
B. Spatio-temporal segmentation
C. Computing the distance from root cells
D. Computing the distance from the borders of spatio-temporal tubes
E. Extraction of cell trajectories

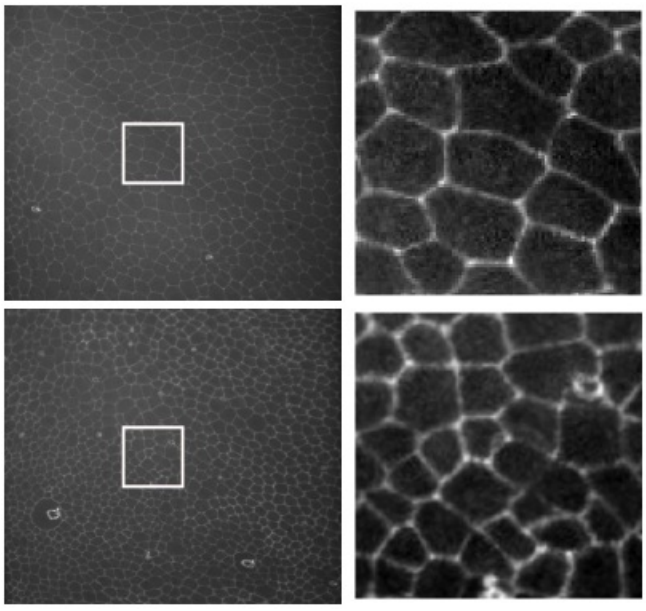

Fig. 1. Data example. In the upper row the 40th frame, in the lower row the 140th frame. Left - whole frame, right - selected 100x100 pixel part of the frame under magnification. White box denotes the selected part.

978-1-4244-4122-8/11/$26.00 ©2011 IEEE

These steps are modular - one can choose different implementation for some of these steps (e.g. cell identification/segmentation), while still taking advantage of our algorithm's performance and robustness for the other steps.

A. Cell identification

Cells in an image are objects with area larger than a certain threshold and smaller than some other threshold. Considering isophotes of the image intensity function, we see that if we approximate the contours of cells with a circle of radius r, this radius lies between some bounds d_1, d_2, $d_1 < r < d_2$. On the other hand, spurious noisy structures are represented by contours of radii significantly less than d_1, $0 < r \ll d_1$. A Level-Set Center Detection (LSCD) algorithm is designed with this property in mind, and we use it to identify cells in an image [2], [3]. In LSCD, we look for a numerical solution to the following equation:

$$u_t + \delta|\nabla u| - \mu|\nabla u|\nabla \cdot \left(\frac{\nabla u}{|\nabla u|}\right) = 0 \qquad (1)$$

where the initial condition is an input image and boundary condition is zero Neumann. δ and μ are the coefficients of advection in the inward normal direction and the curvature regularization, respectively. Function u is defined as $u : R^2 \times [0, T] \to R$. The result of the algorithm is the set of maxima of u at the end of the evolution.

Depending on the signal-to-noise ratio of the image, one should consider filtering of the data in the pre-processing, to remove the image noise. Suitable filter is e.g. Geodesic Mean Curvature Flow (GMCF) smoothing algorithm [4], [5]. However, cell identification, as well as segmentation (see next step), both use a curvature regularization, so they implicitly contain smoothing and the filtering step is not always necessary.

The LSCD method is used separately for every 2D frame of the video. One can see the results of cell identification algorithm in fig. 2.

B. Spatio-temporal segmentation

For segmentation, we use the spatio-temporal Generalized Subjective Surface (GSUBSURF) algorithm [6], [7], [8], [3]. For this algorithm we need first to construct an initial condition. This initial condition should approximate the spatio-temporal tubes according to the information we already have - the better it does, the less time steps of the algorithm we need to perform. We call this initial condition "initial segmentation". For each 2D frame, we create a disc of small radius around each cell identifier and we set pixels inside this disc to value 1, otherwise value stays at 0. GSUBSURF is a numerical solution to this equation:

$$u_t - w_a \nabla g \cdot \nabla u - w_c g|\nabla u|\nabla \cdot \left(\frac{\nabla u}{|\nabla u|}\right) = 0 \qquad (2)$$

where g is an edge detector function, w_a and w_c are advection and curvature parameters of the model. Here, u is defined as $u : R^3 \times [0, T] \to R$. It takes an initial segmentation profile and lets its isosurfaces evolve. We solve this equation in the whole spatio-temporal 3D area.

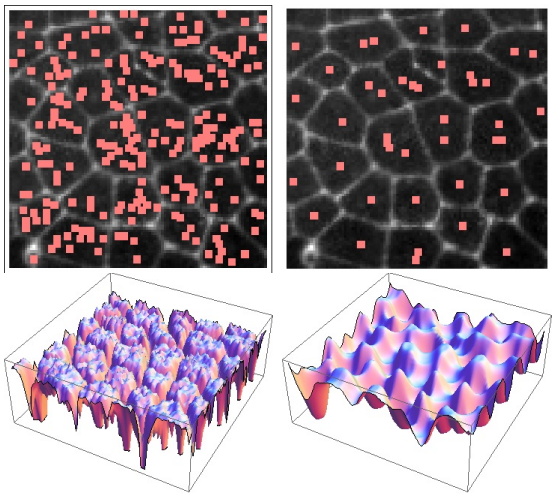

Fig. 2. Cell identification visualization. In the upper row: local maxima of the original image (left) and of LSCD-evolved image (right) are visualized, viewed together with the original image as background. In the lower row: the intensity level functions of the original image (left) and the image after the LSCD evolution (right) are displayed.

During the evolution, shock profiles are created at the inner edges of cells [6], [7], and thus, considering pixels bounded by a specific isosurface, we obtain borders of spatio-temporal 3D tubes representing the result of our segmentation, cf. fig. 3 and 5.

An important property of our spatio-temporal segmentation is that even if a cell identifier in a 2D frame is missing, we can still recover the shape of this cell by segmentation function evolution in the time direction. Furthermore, the spatio-temporal borders of cells are respected in GSUBSURF evolution, so we get separated 3D tubes. Of course, in real data, this separation may not be perfect, but this is solved in later steps of our approach. Both initial and final segmentation can be seen in fig. 3.

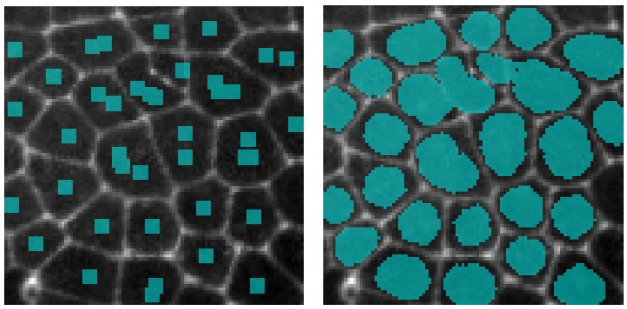

Fig. 3. Spatio-temporal segmentation result visualization. Left, one 2D frame of initial segmentation - small discs created around the cell identifiers. Right, one 2D slice of 3D segmentation result. It is a set of pixels for which the segmentation function has values greater than or equal to the prescribed value.

C. Distance from the root cells

To compute the distance from the root cells, we use the time relaxed eikonal equation, which looks as follows:

$$d_t + |\nabla d| = 1, d(x, t) = 0, x \in \Omega_0 \qquad (3)$$

where d, $d : R^3 \times [0,T] \rightarrow R$, as time increases, approximates the distance from the points where zero Dirichlet condition is prescribed. In this step of the algorithm, Ω_0 is given by the root cell identifiers. We will refer to this distance function as d_1. Technically, this equation is solved as in [9].

If we pick a point in our segmented set of spatio-temporal tubes, we want it to follow a path "down", i. e. in the direction of decrease of this distance function, until it reaches a root cell identifier - see fig. 4. This path is our first naive approach to the trajectory extraction. If the spatio-temporal tubes were perfectly isolated from each other, just following d_1 would be sufficient to get the approximate trajectories.

D. Distance from the borders of the spatio-temporal tubes

Following just d_1 often forces paths to follow borders of the spatio-temporal tubes, rather than their centers - this can be seen in fig. 4, especially in the top left image. This is not the most accurate path. Furthermore, if tubes are not perfectly isolated, paths can slip through holes in borders and give wrong tracking results - see fig. 4, top right image in the figure. The main idea of this step is to force the paths to follow the approximate centers of cells, while following d_1 distance down. Let us define a cell center as a point in cell with maximal distance from border of its spatio-temporal tube. To find this point, we again need to find a distance function.

The distance from the borders of the spatio-temporal tubes is also computed by the eikonal equation. This time, the points with zero Dirichlet condition are the border points of the spatio-temporal tubes. We denote this distance function as d_2.

An important property of this path modification is that if the spatio-temporal tubes meet, no slipping through this meeting point occurs - see fig. 4, in the lower row.

E. Extraction of cell trajectories

For a given point in a 3D spatio-temporal tube, we extract its cell trajectory by minimizing d_1 in a steepest-descent manner while maintaining d_2 maximized. In other words, trajectories go through the spatio-temporal 3D tube, backwards in time, to the root cell identifier, while staying in the cell center in each time frame.

Logically, a cell evolution can be represented as a binary tree. It is rooted at the root cell identifier and it branches out into two children each time the cell divides. As the video starts with many root cells existing already, we should rather talk about a binary forest - forest simply means a set of trees. Tree is constructed in such a way that if we choose a particular node as a representation of a cell, just by following line of its ancestors down to the root, we obtain a trajectory of this cell.

From the data structure point of view, the whole forest consists of nodes. These nodes, besides carrying their temporal and two spatial coordinates, also carry a reference to the parental node, left child node and right child node. As in most of the frames a cell does not divide, we use a standard of always following a left branch of a tree - a right child can be different from NULL if and only if a cell division occured at a given frame. A tree is created in such a way that for a given node we search for a list of predecessors, thus the tracking is computed backwards and the tree is constructed in a top-down manner.

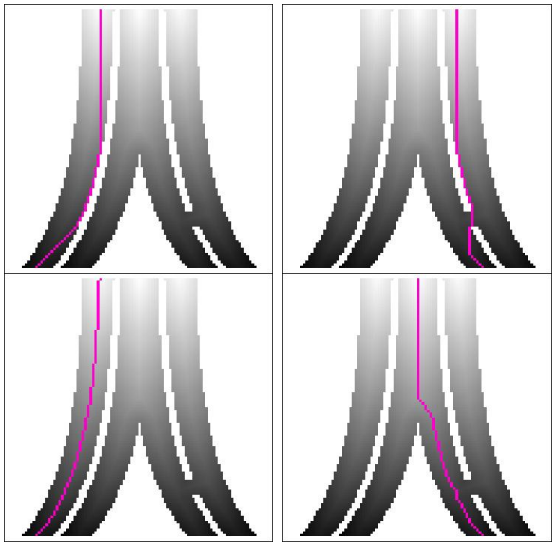

Fig. 4. An illustration of the distance function properties. Image represents 1D + time simulated evolution of the three artificial cells, where the central one divides at about the half time of the evolution - the time axis goes down. There is an artificial "missing-boundary flaw" between the two cells in the lower right part of image. Upper row - trajectories calculated using only distance to the root cells. Tendency to follow the edges rather than the centers (left) and non-robustness against missing boundaries (right) is visible. Lower row - trajectories constructed using both distance function to the root cells and to the borders of tubes. Trajectories tend to follow centers of cells (left) and are robust against missing boundary flaws (right).

In fig. 5 one can see a visualization of spatio-temporal 3D tubes, which were obtained using tracking results.

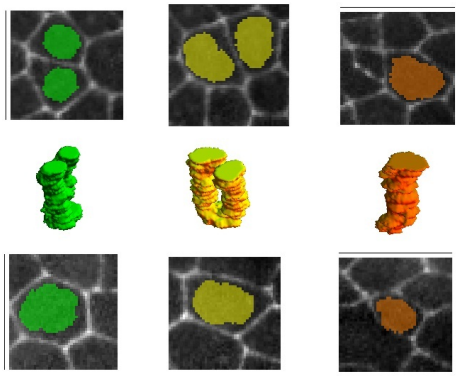

Fig. 5. Visualization of the spatio-temporal 3D tubes. We use tracking results in order to obtain one specific tube from the set of tubes. Time axis points up. "Bifurcated tubes"represent evolution of cells which undergo cell divisions, whereas "simple tubes"represent cells that did not divide.

In order to visualize the tracking results themselves, we assign a color to each cell identified in the beginning of the video. This color is used to identify the cells corresponding to the evolution of the original cells. In fig. 6 and 7 one can see visualization of few frames of tracking results.

III. Discretization and implementation

Equations of GMCF, LSCD and GSUBSURF are discretized and solved using the finite volume method, with pixel/voxel serving as a natural choice of control volume. A detailed discretization procedure is described in [3]. The time-relaxed eikonal equation, used for calculating distance functions, is solved in a way introduced in [9].

IV. Experiments

We have worked with a part of the video covering 100 frames with resolution 100x100 pixels. In the beginning of the video there are 12 cells in the area. These move, mostly to the right, and most of them undergo cell division a few times during the 100 frames. In the end of the video, there are 11 cells corresponding to the original cells identified in the first frame - the others disappear through the right border of the area while moving to the right. Through the left border, some new cells arrive to the area, but we do not track these, as their root cell identifiers are unknown. Using this video, we chose algorithm parameters so that it gives the best possible results. In the LSCD, $\delta = 1.0$ and $\mu = 0.000001$ and we perform 20 time steps. In the spatio-temporal GSUBSURF, setting $w_a = w_c = 0.1$, 100 time steps are performed.

Then, we took two alternative parts of the video, both covering 100 consecutive time steps, both with resolution 100x100 pixels. We wanted to test, how well do the previously set parameters behave under the new conditions. In the first alternative video, there are 16 cells in the beginning and 9 corresponding ones in the end. In the second alternative video there are 13 cells in the beginning, 19 corresponding in the end. Cells in these parts of the video also move to the right.

We measure the success of our approach by counting number of correct and incorrect links between the cell identifiers in two consecutive frames. Number of incorrect links can be understood as the required amount of work to be performed by a user of the software, in order to achieve perfect tracking. We call it a number of "hand-correction" operations.

In the first video, for which the parameters of the model were optimized, we got 1398 correct links out of 1400 total. That means 99.86% success. In the first alternative video, using the previously set parameters, we got 1759 / 1775 - 99.01% success. For the second alternative video we got 1595 / 1600 - 99.69% success.

Tracking results of the original video part can be seen in fig. 6, alternatives are to be seen in fig. 7.

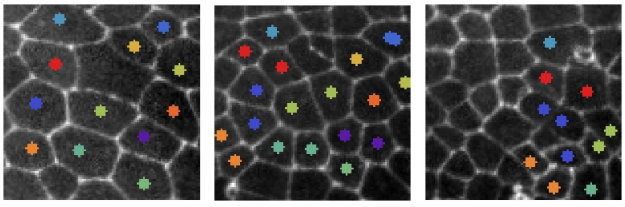

Fig. 6. Tracking in the original video part. From left to right - frame 1, frame 50, frame 100.

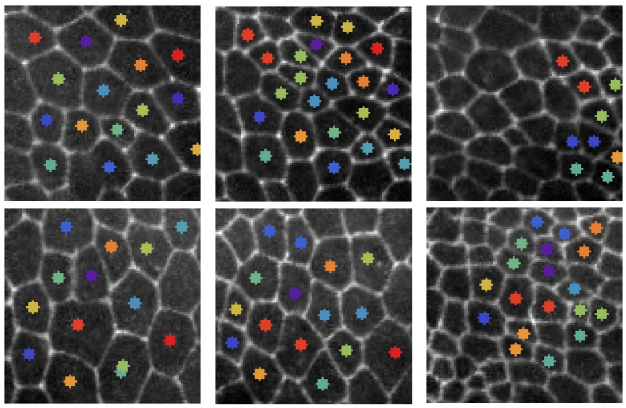

Fig. 7. Tracking in alternative video parts. Upper row - first alternative part, lower row - second alternative part. From left to right (both rows) - frame 1, frame 50, frame 100.

V. Conclusion

In this paper, we have presented an algorithm for tracking cells in image sequences. An algorithm accepts lightly noised input images. It is robust against missing boundaries of cells and missing cell identifiers, thanks to spatio-temporal segmentation. It can overcome imperfections of 3D spatio-temporal tube separation, via combination of two distance functions. Parameters of model, once found, can be used for analysis of similar videos, as we have shown in section about experiments. We have developed this algorithm using a 2D+time video, but its ideas can be extended to 3D+time videos as well.

References

[1] Y. Bellaiche, M. Gho, J. Kaltschmidt, A. Brand, and F. Schweisguth, "Frizzled regulates the localisation of cell-fate determinants and mitotic spindle rotation during asymmetric cell division," *Nature Cell Biology*, vol. 3, pp. 50–57, 2001.

[2] P. Frolkovic, K. Mikula, N. Peyriéras, and A. Sarti, "A counting number of cells and cell segmentation using advection-diffusion equations," *Kybernetika*, vol. 43, no. 6, pp. 817 – 829, 2007.

[3] P. Bourgine, R. Cunderlík, O. Drblíková, K. Mikula, N. Peyriéras, M. Remesíková, B. Rizzi, and A. Sarti, "4D embryogenesis image analysis using pde methods of image processing," *Kybernetika*, vol. 46, no. 2, pp. 226 – 259, 2010.

[4] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *International Journal of Computer Vision*, vol. 22, pp. 61–79, 1997.

[5] Z. Krivá, K. Mikula, N. Peyriéras, B. Rizzi, A. Sarti, and O. Stasová, "3D early embryogenesis image filtering by nonlinear partial differential equations," *Medical Image Analysis*, vol. 14, no. 4, pp. 510 – 526, 2010.

[6] A. Sarti, R. Malladi, and J. Sethian, "Subjective surfaces: A method for completing missing boundaries," *Proceedings of the National Academy of Sciences of the USA*, vol. 97, no. 12, pp. 6258–6263, 2000.

[7] S. Corsaro, K. Mikula, A. Sarti, and F. Sgallari, "Semi-implicit co-volume method in 3D image segmentation," *SIAM Journal on Scientific Computing*, vol. 28, no. 6, pp. 2248 – 2265, 2006.

[8] K. Mikula, N. Peyriéras, M. Remesíková, and A. Sarti, "3D embryogenesis image segmentation by the generalized subjective surface method using the finite volume technique," *Finite Volumes for Complex Applications V: Problems and Perspectives*, pp. 585 – 592, 2008.

[9] P. Bourgine, P. Frolkovic, K. Mikula, N. Peyriéras, and M. Remesíková, "Extraction of the intercellular skeleton from 2D microscope images of early embryogenesis," *Proceeding of the 2nd International Conference on Scale Space and Variational Methods in Computer Vision, Voss, Norway, June 1-5,2009*, pp. 38 – 49, 2009.