**Slovenská technická univerzita**
**v Bratislave**
**Stavebná fakulta**

Študentská vedecká konferencia
Akademický rok 2010/2011

# Nový algoritmus na počítačovú rekonštrukciu vývoja buniek v obrazových postupnostiach morfogenézy.

# A novel algorithm for tracking of cells in morphogenesis image sequences.

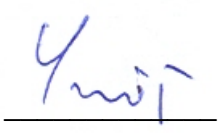
| | |
|---|---|
| Meno Priezvisko študenta: | bc. Michal Smíšek |
| Ročník a program/odbor štúdia: | 2. ročník 2. stupeň., MPM |
| Vedúci práce: | prof. RNDr. Karol Mikula, DrSc. |
| Katedra: | Katedra matematiky a deskriptívnej geometrie |

Bratislava 13. apríl 2011

**Declaration on word of honour:**

I hereby declare that I have worked on this project on my own, using only the resources and literature listed in the bibliography and in correspondence with good scientific practices.

**Acknowledgement:**
I am thankful to my supervisor, Karol Mikula, whose encouragement, guidance and support from the initial to the final level helped me to gain insight into the topic. I also want to thank to Mariana Remešíková, who supported us with valuable suggestions.
I appreciated collaboration with François Graner and Yohanns Bellaïche from Institut Curie, Paris, who provided us with the data and with many inspiring ideas.



Michal Smisek
Bratislava 13th april 2011

**Anotácia:**

V tomto článku popisujeme nový algoritmus na rekonštrukciu vývoja buniek vinnej mušky (drosophila) v procese morfogenézy z mikroskopického videa. Rekonštrukciou vývoja buniek v obrazových postupnostiach rozumieme identifikáciu jednotlivých buniek v obraze a nájdenie ich korešpondencií medzi jednotlivými obrazmi. Hlavnou ideou nášho prístupu je uvažovanie 2D videa ako 3D statického obrazu, ktorý vznikne „naskladaním“ jednotlivých snímok videa na seba. Následne s využitím časopriestorovej metódy subjektívnych povrchov získame informáciu o časopriestorových 3D útvaroch pripomínajúcich trubice, ktoré reprezentujú evolúcie buniek. Nakoniec, s využitím dvoch vypočítaných funkcií vzdialenosti, jednej ako vzdialenosti od kmeňových buniek a druhej ako vzdialenosti od okrajov časopriestorových útvarov, získame trajektórie bunkových evolúcií. Tento prístup možno zovšeobecniť aj na analýzu 3D videa, kde časopriestorové útvary bunkových evolúcií sú 4D objekty.

**Annotation:**

In this article, we present a novel algorithm for tracking of cells in a fruit fly (drosophila) morphogenesis microscopy video. Tracking, in the context of video processing, means identifying logical objects in the frames of the video and finding their correspondences in frame sequences. The main idea of this approach is to consider a 2D + time video as a 3D static image, where frames are stacked atop each other. Using the spatio-temporal generalized subjective surface algorithm we obtain the information about spatio-temporal 3D tubes representing evolutions of cells. Using the two distances, first one as a distance from the root cells and the second one as a distance from the tube boundaries, we obtain the cell trajectories. This approach can be generalized to 3D + time video analysis, where spatio-temporal tubes are 4D objects.

# 1 INTRODUCTION

Cell tracking means extracting spatio-temporal trajectories of cells in a developing organism 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. The basic idea behind this method is seeing a time sequence of 2D images as a 3D image, where separate frames are stacked atop each other.

First, we need to identify cell evolutions as a set of spatio-temporal tubes. We achieve this via spatio-temporal 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 cell identifiers as to the "roots of tubes". Finding a correct trajectory is achieved via computation and use of two constrained distance functions.

# 2 ALGORITHM STEPS

We have at disposal a video of cellular structure of drosophila in morphogenesis stage. Video displays cell membranes. Data is saved in .TIF file format. Video consists of 199 frames and has resolution 569 x 500 pixels. File uses 8 bit encoding per pixel, so pixel intensity can take on 256 different values, ranging from 0 to 255. Video was obtained using a confocal laser microscope. In fig. 1 one can see the visualization of image data.

Our algorithm consists of these steps:

- Image filtering
- Cell identification
- Spatio-temporal segmentation
- Computing the distance from root cells
- Computing the distance from the borders of spatio-temporal tubes
- Extraction of cell trajectories

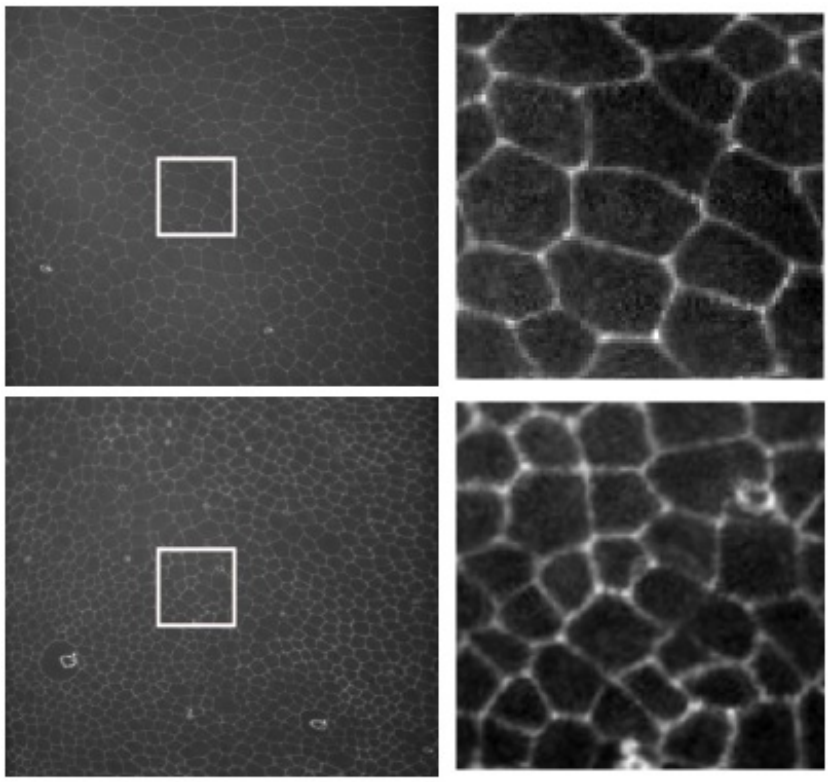


Figure 1: Data example. In the upper row the 40th frame, in the lower row the 140th frame. Left - whole frame, right - selected 100x100 pixel part of the frame under magnification. White box denotes the selected part.

## 2.1 Image filtering

Depending on the signal-to-noise ratio of the image, one should consider filtering of the data, to remove the image noise. For this purpose, we use the Geodesic Mean Curvature Flow (GMCF) smoothing algorithm[7]. GMCF is capable of filtering the image noise while preserving edges of the image[3, 6, 10, 7]. In GMCF, one solves the following equation:

$$u_t - |\nabla u| \nabla \cdot \left( g \frac{\nabla u}{|\nabla u|} \right) = 0 \tag{1}$$

where $u$ is the solution of the problem, $u_t$ is the time derivative and $g = g\left(|\nabla G_\sigma * u|\right)$ is an edge detector function. The initial condition is the input image and we consider zero Neumann boundary condition.

It is worth noting that the two further steps of our approach, cell identification and segmentation, both use a curvature regularization. They im-

plicitly contain smoothing, so the filtering step is not always necessary in our data processing. On the other hand, we use the filtered images for correcting the cell identification results - see subsection 2.2 for more details.

## 2.2 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[5, 1]. In LSCD, we look for a numerical solution to the following equation:

$$u_t + \delta|\nabla u| + \mu|\nabla u|\nabla \cdot \left(\frac{\nabla u}{|\nabla u|}\right) = 0 \tag{2}$$

where the initial condition is an input image and boundary condition is zero Neumann. $\delta$ and $\mu$ are the coefficients of advection in the inward normal direction and the curvature regularization, respectively. The result of the algorithm is the set of maxima of $u$ at the end of the evolution. The LSCD algorithm sometimes introduces spurious cell identifiers on the edges of cells - these can be removed by detecting large differences in filtered image, in small areas around the identifiers.

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.

## 2.3 Spatio-temporal segmentation

For segmentation, we use the spatio-temporal Generalized Subjective Surface (GSUBSURF) algorithm[9, 4, 8, 1]. 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". In each 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:

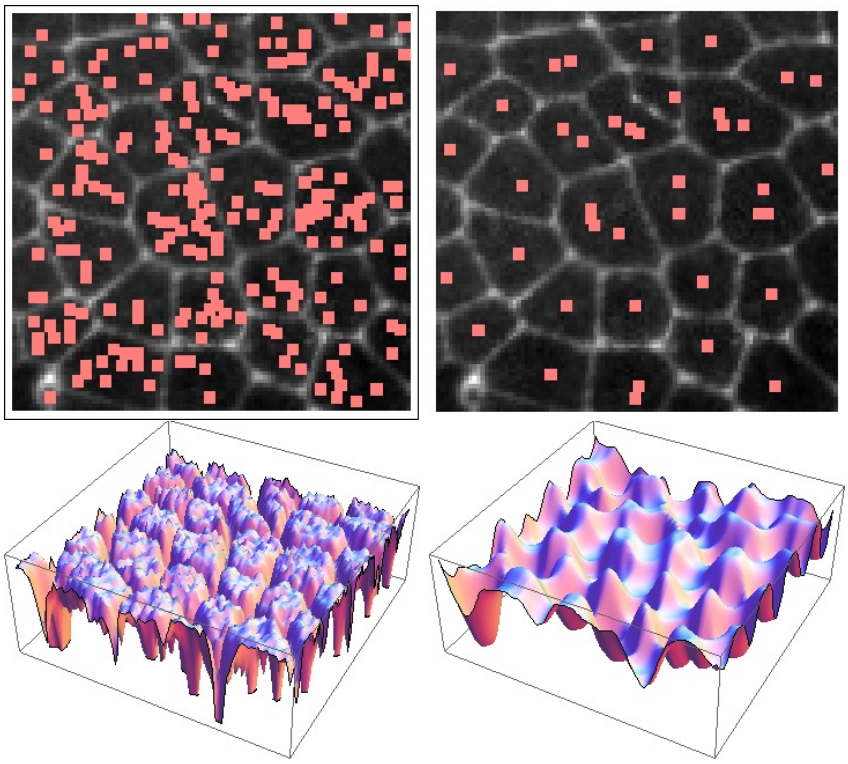


Figure 2: Cell identification visualization. In the upper row: local maxima of the original image (left) and of LSCD-evolved image (right) are visualized, viewed together with the original image as background. In the lower row: the intensity level functions of the original image (left) and the image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

$$u_t - w_a \nabla g \cdot \nabla u - w_c g |\nabla u| \nabla \cdot \left( \frac{\nabla u}{|\nabla u|} \right) = 0 \tag{3}$$

where $g$ is an edge detector function, $w_a$ and $w_c$ are advection and curvature parameters of the model. It takes an initial segmentation profile and lets its isosurfaces evolve.

At the end of the evolution, we use a selected isosurface as a border between a set of spatio-temporal 3D tubes and the rest of the image.

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.

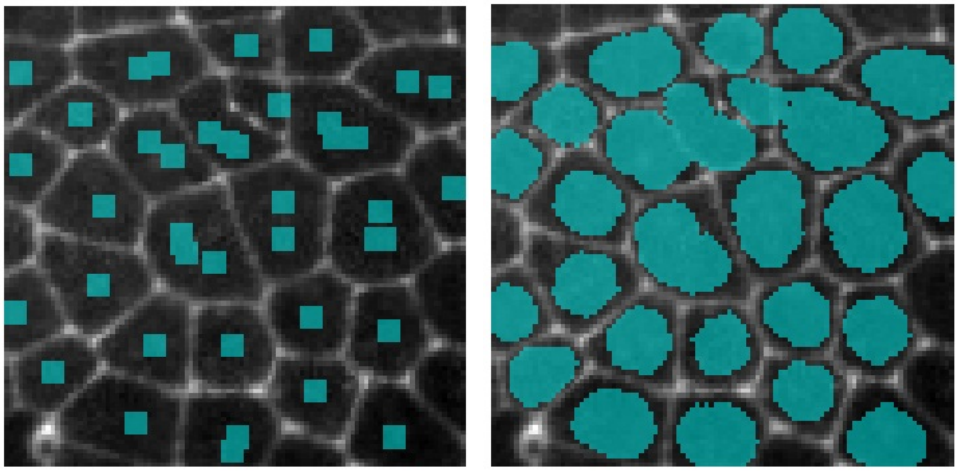


Figure 3: Segmentation result visualization in one frame. Left, initial segmentation - small discs created around the cell identifiers. Right, the segmentation result. It is a set of pixels for which the segmentation function has values greater than or equal to the prescribed value.

## 2.4 Distance from the root cells

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

$$d_t + |\nabla d| = 1, d(x,t) = 0, x \in \Omega_0 \tag{4}$$

where $d$, as time increases, approximates the distance from the points where zero Dirichlet condition is prescribed. In this step of the algorithm, $\Omega_0$ is given by the root cell identifiers. We will refer to this distance function as $d_1$. Technically, this equation is solved as in [2].

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.

## 2.5 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.

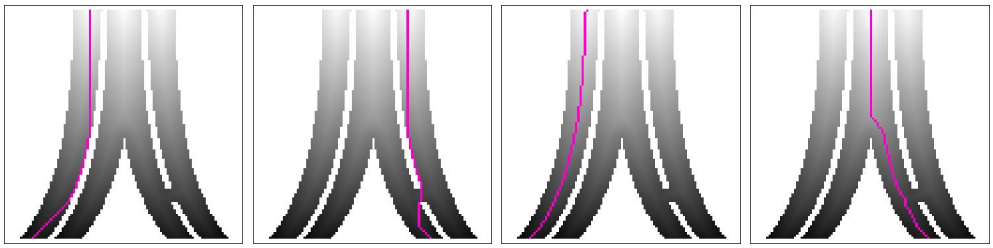


Figure 4: An illustration of the distance function properties. Image represents simulated evolution of the three artificial cells, where the cental one divides at about the half time of the evolution - the time axis goes down. There is an artificial "missing-boundary flaw" between the two cells in the 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 boudary flaws (right).

### 2.6 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 doesn't 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.

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

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.

## 3 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 [1]. The time-relaxed eikonal equation, used for calculating distance functions, is solved in a way introduced in [2].

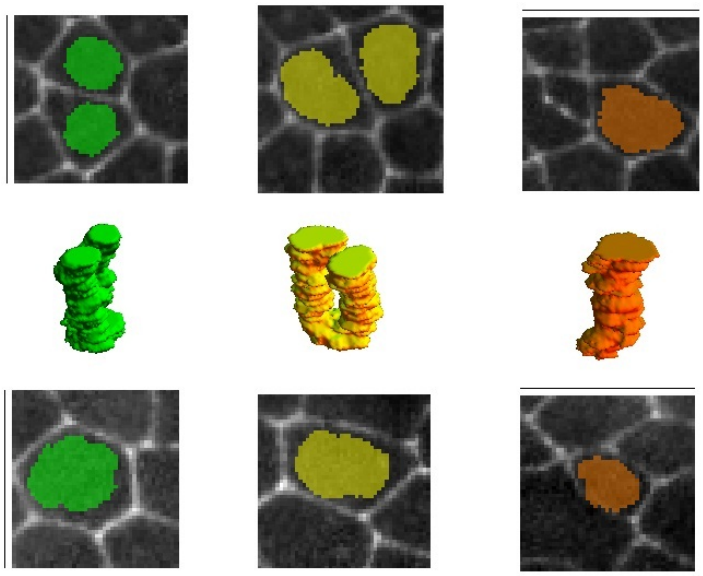


Figure 5: Visualization of the spatio-temporal 3D tubes. We use tracking results in order to obtain a specific tube from a set of tubes.

# 4 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.

For the GMCF, we use the edge detector function $g(s) = 1/(1 + Ks^2)$, $K = 2000$, taking 100 time steps. 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.

Computing the whole sequence of algorithm steps took approximately 30 minutes on Intel Core 2 Duo T7250, 2.00 GHz. Tracking results of the original video part can be seen in fig. 6, alternatives are to be seen in fig. 7.

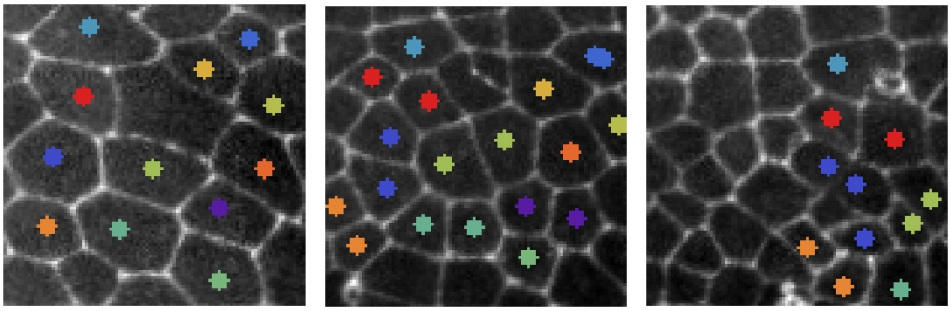


Figure 6: Tracking in the original video part. From left to right - frame 1, frame 50, frame 100.

# 5 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.

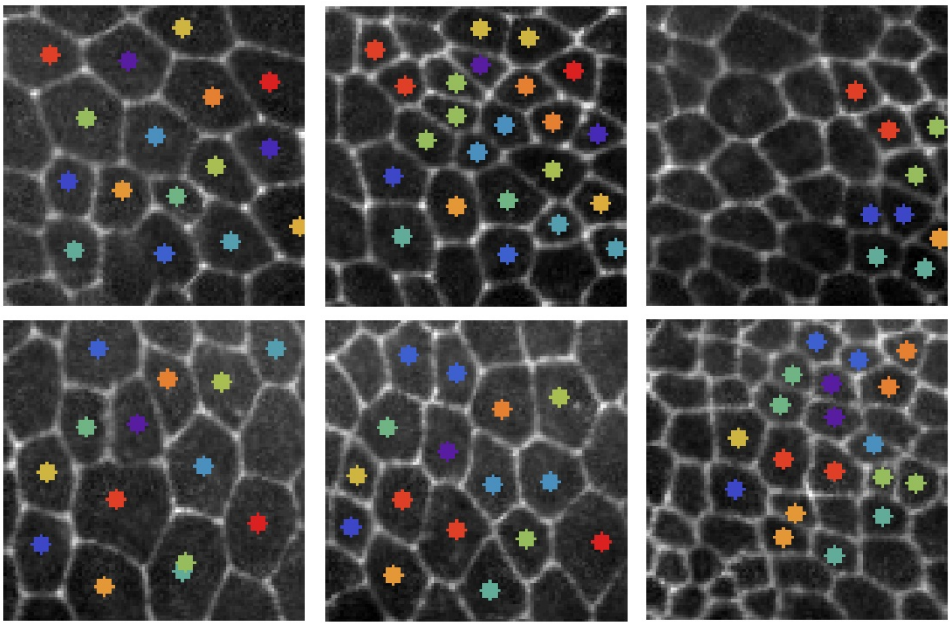


Figure 7: Tracking in alternative video parts. Upper row - first alternative part, lower row - second alternative part. From left to right (both rows) - frame 1, frame 50, frame 100.

# References


[1] 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*, 46(2):226 – 259, 2010.

[2] 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*, pages 38 – 49, 2009.

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

[4] S. Corsaro, K. Mikula, A. Sarti, and F. Sgallari. Semi-implicit co-volume method in 3D image segmentation. *SIAM Journal on Scientific Computing*, 28(6):2248 – 2265, 2006.

[5] P. Frolkovic, K. Mikula, N. Peyriéras, and A. Sarti. A counting number of cells and cell segmentation using advection-diffusion equations. *Kybernetika*, 43(6):817 – 829, 2007.

[6] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and Yezzi A. Conformal curvature flows: From phase transitions to active vision. *Archive for Rational Mechanics and Analysis*, 134:275–301, 1996.

[7] 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*, 14(4):510 – 526, 2010.

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

[9] A. Sarti, R. Malladi, and J. Sethian. Subjective surfaces: A method for completing missing boundaries. *Proceedings of the National Academy of Sciences of the United States of America*, 97(12):6258–6263, 2000.

[10] C. Yunmei, B. C. Vemuri, and L. Wang. Image denoising and segmentation via nonlinear diffusion. *Computers & Mathematics with Applications*, 39(5-6):131 – 149, 2000.