NUMERICAL ALGORITHM FOR TRACKING CELL DYNAMICS IN 4D BIOMEDICAL IMAGES *

KAROL MIKULA †, NADINE PEYRIÉRAS ‡, AND RÓBERT ŠPIR §

Abstract. The paper presents new numerical algorithm for an automated cell tracking from large-scale 3D+time two-photon laser scanning microscopy images of early stages of zebrafish (Danio rerio) embryo development. The cell trajectories are extracted as centered paths inside segmented spatio-temporal tree structures representing cell movements and divisions. Such paths are found by using a suitably designed and computed constrained distance functions and by a backtracking in steepest descent direction of a potential field based on these distance functions combination. The naturally parallelizable discretization of the eikonal equation which is used for computing distance functions is given and results of the tracking method for real 4D image data are presented and discussed.

1. Introduction. In this paper we present a new method for tracking cells in 4D biomedical images (3D+time image sequences). The method is based on extraction of the cell trajectories as centered paths inside 4D spatio-temporal tree structures obtained by segmentation of 4D images. Our method thus further develops ideas given in [1, 10]. In the presented approach, the 4D segmentation is obtained by creating a spatio-temporal tubes around the cell identifiers given as a result of suitable image filtering [8, 5, 4, 7] followed by a cell detection algorithm [6, 2, 11]. Then a computation of constrained distance functions inside 4D segmentation is performed by solving numerically a spatially 4D eikonal equation. Since this is a large-scale computational problem in case of real 4D image data, the parallel implementation is necessary and thus developed. By a proper combination of computed distance functions we build a potential field which is backtracked in steepest descent direction in order to get the cell trajectories. Consequently, the cell lineage tree can be constructed by detecting merging trajectories when going backward in time indicating mitosis and thus a branching node of the cell lineage tree.

Our work is motivated by the recent research in biology and medicine where the reconstruction of cell population dynamics is crucial for obtaining the cell lineage trees and study of formation and evolution of morphogenetic structures. Such research is related to embryonic development of organisms as well as to anticancer drug design.

Among other imaging technologies, the multi-photon laser scanning microscopy is able to deliver in-vivo 3D+time image sequences of long periods of the embryonic development. Due to a similarity with human in many aspects and due to transparency for the microscope, the zebrafish (Danio rerio) is studied extensively and strong demand for reliable cell tracking methods exists [13]. Although a large amount of work has been already done, e.g. by a combination of various image processing techniques and manual inspection, for the zebrafish developmental stages up to about one thousand cells [13], there is a great challenge to study embryogenesis at very complex stages of development with thousands of cells present. The tracking of cells and construction of the cell lineage tree for such complex stages of embryogenesis is

*This work was supported by the grant APVV-0184-10

†Department of Mathematics, Slovak University of Technology, Radlinskeho 11, 813 68 Bratislava, Slovakia

‡Institut de Neurobiologie Alfred Fessard, CNRS UPR 3294 Av. de la Terrasse, 91198 Gif-sur-Yvette, France

§Department of Mathematics, Slovak University of Technology, Radlinskeho 11, 813 68 Bratislava, Slovakia

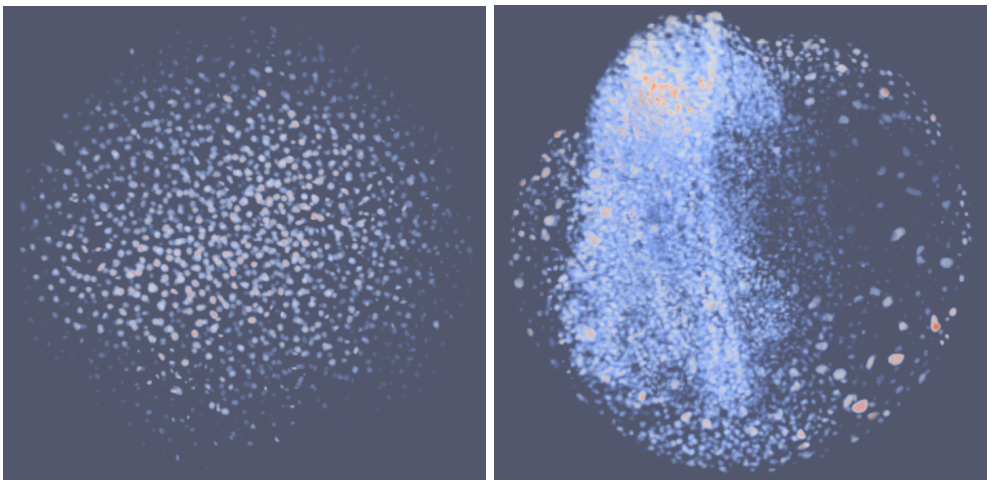

FIG. 1.1. *Left: volume rendering of the cell nuclei data in the first 3D image (4 hours after fertilization). Right: volume rendering of the cell nuclei data in the last 3D image (12 hours after fertilization).*

a very difficult problem unsolved satisfactorily yet. A seminal work towards building the cell lineage tree for the complex stages of zebrafish embryo development based on optic flow estimation followed by a stochastic simulated annealing minimization of a heuristic energy functional has been done in [9]. After the construction of the tree, the individual cells or cell populations can be tracked. In contrast to [9], in our approach we first extract all possible cell trajectories inside the 3D+time image data set and then the cell lineage tree can be reconstructed by finding merging trajectories going backward in time.

Our data for the cell tracking come from two-photon laser scanning microscopy and represent the first hours of zebrafish embryo development, approximately from the 4th until 10th-20th hour. The labeling of cell nuclei (and cell membranes) is obtained by expression of the fluorescence protein through its RNA injection performed at the one-cell stage. The 3D images are obtained by moving the focal plane from the top more deeply inside the embryo and their quality depends on the speed of scanning in one plane. Many various quality datasets are available thanks to EC Embryomics and BioEmergences projects (http://bioemergences.iscpif.fr/bioemergences/). The data quality is related to a level of noise depending mainly on the size of the time step between acquisition of subsequent 3D images. The acquisition step ranges from 50 seconds to 5 minutes. A longer time step produces better 3D image quality, one has a good visual impression of what is happening during embryogenesis and such data is well suited for segmentation purposes e.g. for obtaining a shape of cells and other their characteristics during the embryogenesis [16, 11]. On the other hand, such data is not suitable for tracking since the cells move too far between single 3D images and consequently mother-daughters cell correspondences can be lost. In Figs. 1.1 and 1.2 we plot an example of embryo development at the beginning and at the end of the imaging. In Fig. 1.1 we see visualization of 3D cell nuclei images which are used in our tracking method. Although the nuclei images contain certain level of noise when using time step about one minute they are suitable for application of the presented tracking

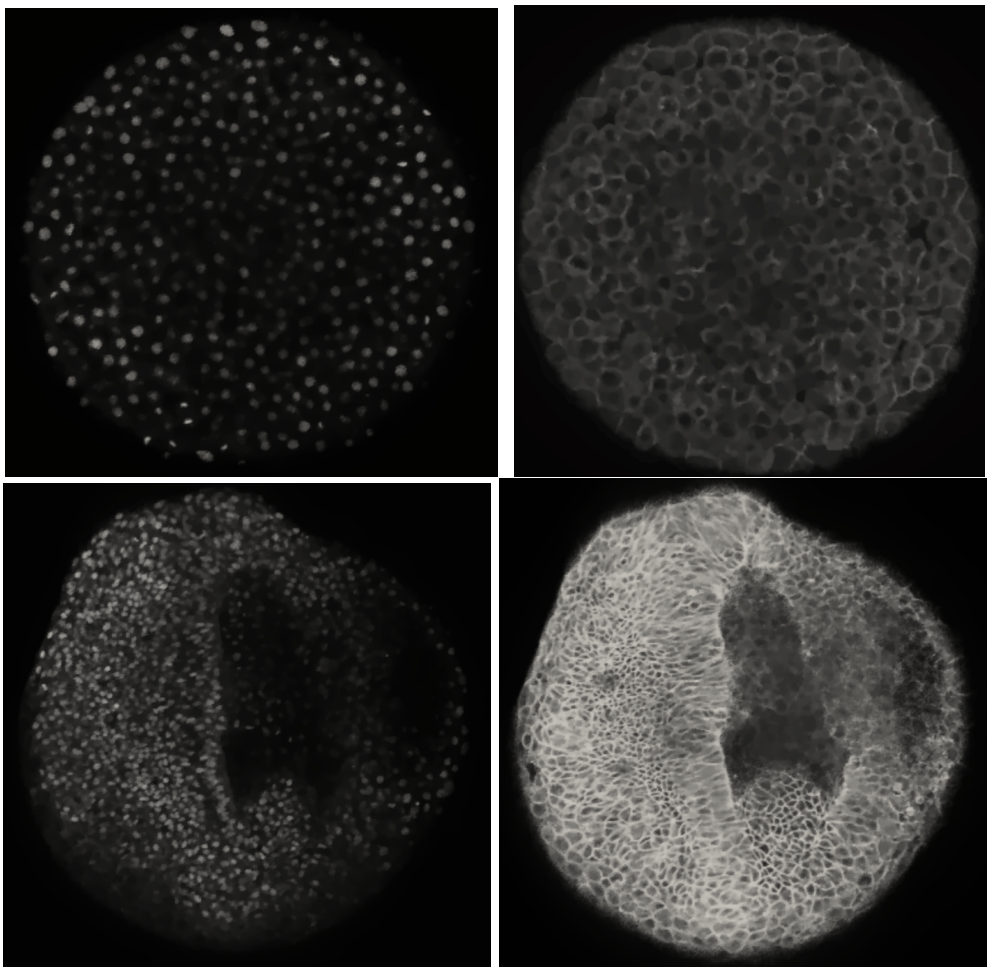

FIG. 1.2. *Top left: 2D slice of raw data 4 hours after fertilization (cell nuclei), top right: 2D slice of raw data 4 hours after fertilization (cell membranes). Bottom left: 2D slice of raw data 12 hours after fertilization (cell nuclei), bottom right: 2D slice of raw data 12 hours after fertilization (cell membranes).*

technique. In Fig. 1.2 one can clearly observe how the zebrafish embryo grows from an unorganized set of cells to complex stages of development containing presumptive organs of the future zebrafish adult. Here we show also the cell membane images in order to see differentiation of cells after 12 hours of embryo development. However, the overall quality of the 3D membranes data is still not suitable for application of our tracking method although it would be natural and desirable, cf. [1].

The paper is organized as follows. In the next section we present our approach to the trajectories extraction and discuss numerical approaches used in the tracking method. Then we discuss numerical experiments devoted to testing the tracking method as well as we present processing of the real 3D+time image sequence of the early zebrafish brain embryogenesis.

2. The cell tracking algorithm. Our method for the cell trajectories extraction is composed of the following steps:

- construction of a 4D segmentation yielding the 4D spatio-temporal tree structure inside which two (constrained) distance functions will be computed,
- computation of the first constrained distance function D giving distance of any point to the most far cell identifier to which it is continuously connected inside the 4D segmentation,
- computation of the second constrained distance function D_B giving distance of any point to the boundary of the 4D segmentation,
- building a potential field V for tracking by using a suitable combination of two computed distance functions,
- extraction of the steepest descent paths of the potential field,
- centering the extracted paths inside the 4D spatio-temporal trees in order to get the cell trajectories

We note that after successful trajectories extraction the reconstruction of the cell lineage tree can be performed by detecting trajectories which merge together when going backward in time indicating mitosis and thus a branching node of the cell lineage tree.

From mathematical point of view, the 3D+time image sequence is understood as a function $u(x_1, x_2, x_3, \theta)$, $u : \Lambda \to [0, 1]$, where Λ is a bounded spatio-temporal (rectangular) subdomain of R^4, (x_1, x_2, x_3) is a spatial point and θ represents a time.

2.1. Building the 4D segmentation. We call 4D segmentation a spatio-temporal structure which approximates the space-time movement of cell nuclei in 3D+time image sequence. Due to [16, 2, 11] the shape of cell nuclei during zebrafish embryogenesis is reasonably approximated by spheres or ellipsoids. Thus, in order to construct 4D segmentation we use cell identifiers [6, 2, 11] detected in all time steps, $s^l_m, m = 1, \ldots, n^l_C, l = 1, \ldots, N_\theta$ (m denotes cell identifier index at time step l and N_θ is number of time steps), and create 4D ellipsoids around all these points. Currently we are using the same halfaxes in space equal to $2dx$, where $dx = dx_1 = dx_2 = dx_3$ areis the voxel size (same in all three spatial dimensions) and with halfaxis equal to $d\theta$ in temporal direction. The nonzero temporal halfaxis is important due to the time overlap which we create and thus we improve connectivity of 4D spatio-temporal tree structures. Thanks to the time overlap we also make connected such branches of the 4D spatio-temporal tree where a cell center is not detected in one frame but it is detected in two neighboring frames and thus we correct false negative errors of the cell center detection algorithm.

We also have to note that above approach is not optimal in the sense of getting a real 4D image segmentation of cell nuclei movement. A more realistic approach, using 4D generalized subjective surface (GSUBSURF) method [15, 4, 7], was developed in [10]. Such method, however, can currently be used only for short time sequences and smaller portions of full 3D volume due to its high computational complexity. Utilization of GSUBSURF method for long 3D+time image sequences remains a great challenge from modeling, discretization and computational point of view and it will be an objective of our further study. On the other hand, the simplified approach suggested and used in this paper also gives, thanks to the time overlap, the reliable 4D segmentation which can be used in further steps of cell trajectories extraction.

As outlined above we have to compute distance functions inside the 4D segmentation (we called them constrained because all the computations are performed only inside it) in order to build the potential field for tracking. So now we present the

numerical scheme which is used in the distance functions computation.

2.2. 4D Rouy-Tourin scheme with fixing for computing distance function. In order to compute distance function we solve numerically the so-called "time" relaxed eikonal equation

$$d_t + |\nabla d| = 1 \tag{2.1}$$

for the unknown function $d(x_1, x_2, x_3, \theta, t)$. We solve here spatially 4D problem, so ∇d is a 4D gradient of the function d, i.e. the vector of partial derivatives with respect to x_1, x_2, x_3 and θ variables. The distance function is obtained as an equilibrium of the numerical solution, i.e. it satisfies numerically the classical eikonal equation $|\nabla d| = 1$. For the equation (2.1), we have to prescribe zero Dirichlet condition on the set from which we compute the distances. In our cases, we will prescribe the zero values either for a particular set of cell identifiers or for the boundary points of the 4D segmentation.

We denote the 4D doxels by V_{ijkl}. Without losing generality, we rescale the time step $d\theta$ to be equal to $dx_1 = dx_2 = dx_3$ and denote their common value by h_D (standardly we set $h_D = 1$). Let d^n_{ijkl} denote the approximate value of solution d in barycenter of V_{ijkl} in a discrete step $t^n = n\tau_D$ where τ_D is the length of step discretizing t variable. Then, for every V_{ijkl} we define the index set N_{ijkl} of all (p, q, r, s) such that $p, q, r, s \in \{-1, 0, 1\}$, $|p| + |q| + |r| + |s| = 1$. In order to built the scheme, for any $(p, q, r, s) \in N_{ijkl}$, we define

$$D^{pqrs}_{ijkl} = \left(\min\left(d^{n-1}_{i+p,j+q,k+r,l+s} - d^{n-1}_{ijkl}, 0\right)\right)^2 \tag{2.2}$$

and then also

$$\begin{aligned} M^{1000}_{ijkl} &= \max\left(D^{-1,0,0,0}_{ijkl}, D^{1,0,0,0}_{ijkl}\right),\ M^{0100}_{ijkl} = \max\left(D^{0,-1,0,0}_{ijkl}, D^{0,1,0,0}_{ijk}\right) \\ M^{0010}_{ijkl} &= \max\left(D^{0,0,-1,0}_{ijkl}, D^{0,0,1,0}_{ijkl}\right),\ M^{0001}_{ijkl} = \max\left(D^{0,0,0,-1}_{ijkl}, D^{0,0,0,1}_{ijkl}\right). \end{aligned} \tag{2.3}$$

Using this notations, the 4D Rouy-Tourin scheme [14] for solving equation (2.1) has the following form

$$d^n_{ijkl} = d^{n-1}_{ijkl} + \tau_D - \frac{\tau_D}{h_D}\sqrt{M^{1000}_{ijkl} + M^{0100}_{ijkl} + M^{0010}_{ijkl} + M^{0001}_{ijkl}}\,. \tag{2.4}$$

Due to stability reasons, the coupling $\tau_D = h_D/2$ is used.

From (2.2) one can see that numerical solution given by the Rouy-Tourin scheme (2.4) depends only on neighours with lower values. The steady state of numerical solution in any doxel is thus reached only after stabilization of values in all neighbouring doxels with lower values. Moreover, if the steady state is reached, it cannot be changed later by any influence of other neighbours. It shows that the steady state of numerical solution, representing desired distance function, is achieved subsequently from smaller to bigger values and allow us to fix values in doxels which are already in equilibrium and omit further computations in such doxels. This fixing strategy was introduced in [3, 11] in order to speed-up the basic method (2.4) and it will be significantly utilized here in building the first distance function D as will be seen in the next subsection.

Let us consider the index set $\mathcal{I}$ of all indices (i, j, k, l) and the set $\mathcal{F}^n$ that contains the indices $(i, j, k, l) \in \mathcal{I}$ of doxels where the steady state has been already reached, i.e.

where $|d^n_{ijkl} - d^{n-1}_{ijkl}| < \varepsilon$, where ε is a given threshold (in all presented computations we use $\varepsilon = 0.01$). The set $\mathcal{F}^0$ contains indices of all doxels which have given values and are fixed initially. Since we perform all computations only in the doxels that have not yet reached the steady state and the number of fixed doxels is gradually increasing, the computational time needed in one time step of (2.4) is decreasing and finally all doxels are fixed. Here is our algorithm of Rouy-Tourin scheme with the fixing:

- Do while $\mathcal{F}^n \neq \mathcal{I}$
- Do for all $(i,j,k,l) \in \mathcal{I}$
  - if $(i,j,k,l) \in \mathcal{F}^n$ then continue
  - else
    - update d^{n+1}_{ijkl} using (2.4)
    - if $d^{n+1}_{ijkl} = d^n_{ijkl}$ then $\mathcal{F}^n \cup \{(i,j,k,l)\}$
- $n = n + 1$

The 4D Rouy-Tourin scheme with the fixing was parallelized using OpenMP programming interface for shared memory parallel servers. The OpenMP parallelization was chosen in order to eliminate the need for interprocess communication. If MPI would be used for our purposes, around 100MB would have to be sent and received between neighboring parallel processes in each step of distance function computation. Clearly, such communication can make overall parallel implementation inefficient. On the other hand, when using OpenMP we have to run the computation on a single shared memory computer and we are limited by the amount of available memory of such server. E.g., for the numerical experiment with 479 time step dataset with $512 \times 512 \times 104$ voxels in each 3D volume, discussed in section 3, we needed roughly 140 GB of shared memory. Since there are currently quite standard parallel servers with 32 processors and 256GB of shared memory available, we can use our OpenMP parallel approach for the long-time image sequence processing. Moreover, we obtained the speed-up 23.5 for 32 threads by using our OpenMP parallel implementation. Such speed-up was obtained thanks to utilization of the NUMA (non-uniform memory access) library functions which allow the optimal NUMA nodes memory usage on the nowadays standard servers with NUMA architecture.

2.3. Building the potential field for tracking. As described in the beginning of this section, for building the potential field V we compute two types of distance functions, D and D_B, inside the 4D spatio-temporal tree structures. We call them constrained distance functions because all the calculations are constrained by the boundaries of the 4D segmentation. Due to that fact, the computed distances between doxels of the 3D+time image sequence are not a standard Euclidean distances in R^4 but they represent a minimal Euclidean paths between the points inside the 4D segmentation. The 4D segmentation containing 4D spatio-temporal tree structures can be represented by a 4D piecewise constant function, with some BIG value (which is arbitrary but bigger than any distance which can be obtained inside the 4D segmentation) outside of the segmentation and with zero value inside it.

The first constrained distance function $D(x_1, x_2, x_3, \theta)$ is built subsequently by the following approach:

- Initialize D by setting $D(x_1, x_2, x_3, \theta) = BIG$ for all doxels outside of the 4D segmentation and by fixing all these doxels, and setting $D(x_1, x_2, x_3, \theta) = 0$ for all doxels inside the 4D segmentation. Let us remind that no further computations are performed in fixed doxels. Then

• for every $l = 1, 2, \ldots, N_\theta - 1$

• take all doxels, corresponding to cell identifiers $s_m^l, m = 1, \ldots, n_C^l$ at the lth step, where the value is not fixed yet. Fix the value in these doxels (it was set to 0 in initialization or in previous step $l-1$) and compute distance $D(x_1, x_2, x_3, \theta)$ to these cell identifiers in all doxels which are not yet fixed by using the scheme described in the previous subsection,

• during the computation of $D(x_1, x_2, x_3, \theta)$ at the lth step the total number of unfixed doxels is gradually decreasing. If the number of unfixed doxels stops to decrease we have already computed correct distances in all simply connected components of the 4D segmentation reachable from the l-th step cell identifiers and all such doxels are fixed. After that the Rouy-Tourin scheme would just uniformly increase values (by τ_D in every next step n) in all so far unfixed doxels belonging to components of 4D segmentation which are not reachable from any cell identifier s_m^l. So, we stop the computation at step l, set $D(x_1, x_2, x_3, \theta) = 0$ in all remaining unfixed doxels and go to step $l + 1$.

At the end all doxels inside the 4D segmentation are fixed by the value of distance to the most far cell identifier to which it is continuosly connected. In Fig. 2.1 we show how the distance function D contains information about simply connected components of the 4D segmentation. Inside the regions encompassed by the BIG values, the value of D is growing from zero, in cell identifier where any simply connected component begins, up to a maximal value, where the simply connected component ends. Such construction of the distance function D allows us to get all, also partial, cell trajectories contained in our 4D spatio-temporal tree structure. Its utilization represents an important novelty with respect to [1, 10] where just distance to the centers detected in the first time step was used, so only the cell trajectories starting in the first 3D image could be detected.

To update D in a single doxel in each time step of the Rouy-Tourin scheme we have to perform evaluation of expressions (2.2)-(2.3) for eight neighbouring doxels. Due to advective character of equation (2.1) and due to stability condition, we need roughly twice as many time steps as is the number of the 3D volumes to obtain the steady state of numerical solution. So, if we performed the distance function computations in all full 3D volumes, the computational complexity would be roughly estimated as $8 \times (2 \times N_\theta) \times N_1 \times N_2 \times N_3 \times N_\theta$, where N_1, N_2, N_3, N_θ represents the dimensions in spatial and temporal directions. Fortunately, we need to calculate D only inside the 4D segmentation, which represents only a small portion of the 4D image. In our real data set with 479 3D volumes the doxels inside the 4D segmentation represent only 3.5% of the whole 4D image. This reduces the calculation complexity considerably. Moreover, 98.8% of doxels inside 4D segmentation was fixed in the first step $l = 1$. The computation of D took 4600 seconds in $l = 1$ and only 200 seconds for $l = 2, \ldots, N_\theta - 1$ altogether.

Now, looking at Fig. 2.1 and also Fig. 2.2 left, we can think about D as a potential field and traverse it in the steepest descent direction from the highest value in every simply connected component to the zero value. The paths obtained in such way would represent good approximation of the space-time cell trajectories. And, if the 4D segmentation would contain only perfectly separated 4D spatio-temporal tree structures, we would obtain correctly all (also partial) cell trajectories which can be extracted from the data. Unfortunately, in the real 4D data it is not always the case and we must deal with imperfections given mainly by a cells overlapping. In order to overcome this difficulty we have to keep the extracted paths in a certain distance from

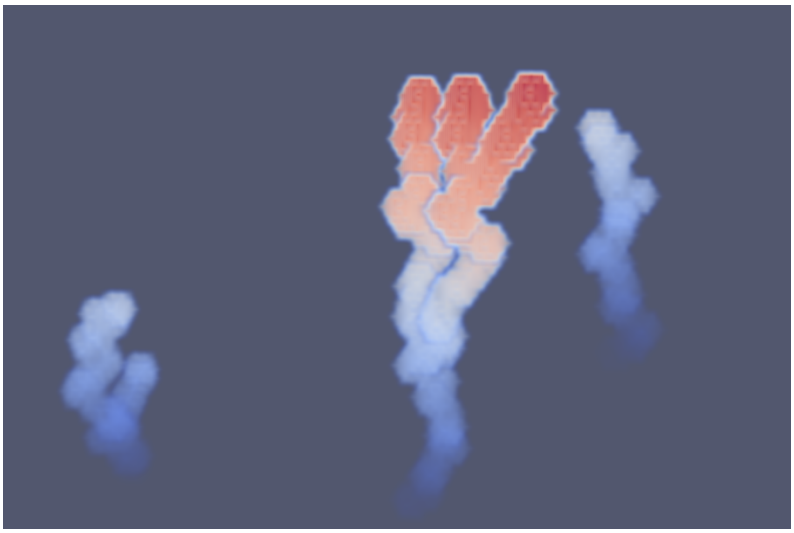

FIG. 2.1. *Plot of the constrained distance function D showing its values in various simply connected components.*

the spatio-temporal cell boundaries or, in other words, they should be more centered inside the 4D segmentation.

This can be achieved by using the constrained distance function $D_B(x_1, x_2, x_3, \theta)$ [12, 1, 10] values of which grow from boundaries to the center of the 4D spatio-temporal trees, see Fig. 2.2 right. For the computation of D_B we use again the Rouy-Tourin scheme with fixing, but we set $D_B(x_1, x_2, x_3, \theta) = 0$ and fix it for all (x_1, x_2, x_3, θ) outside and on the boundary of the 4D segmentation.

Based on these observations we build a potential field

$$V(x_1, x_2, x_3, \theta) = D(x_1, x_2, x_3, \theta) - D_B(x_1, x_2, x_3, \theta) \tag{2.5}$$

which is used in our algorithm for the extraction of cell trajectories.

2.4. Extraction of the cell trajectories. The cell trajectory will be represented by a series of points in space-time (discrete spatio-temporal curve) for which we prescribe the condition that there is exactly one point in every time step $l = N_b, \ldots, N_e$, $1 \leq N_b < N_e \leq N_\theta$. The extraction of cell trajectories is realized in two steps

- first, we use backtracking in time by the steepest descent direction of the potential V built in (2.5) starting from all cell identifiers $s_m^l, m = 1, \ldots, n_C^l$ detected in all time steps $l = N_\theta, \ldots, 2$,
- then, we center all the extracted paths inside the 4D spatio-temporal trees by using constrained distance function D_B in order to eliminate duplicates and thus to obtain the cell trajectories.

The first step is realized as follows: Let s_m^l be one of the cell identifiers detected in the lth time step. Let us define a temporary point $P_T^l = s_m^l$. Then, we search recursively in the nearest vicinity of P_T^l, but only in the current time step l and previous time step $l-1$, for a doxel with the minimal value of the potential V which is also strictly less than the value of potential at the temporary point. The extracted path point P^l for the time step l is defined as the (last in search) doxel from which we

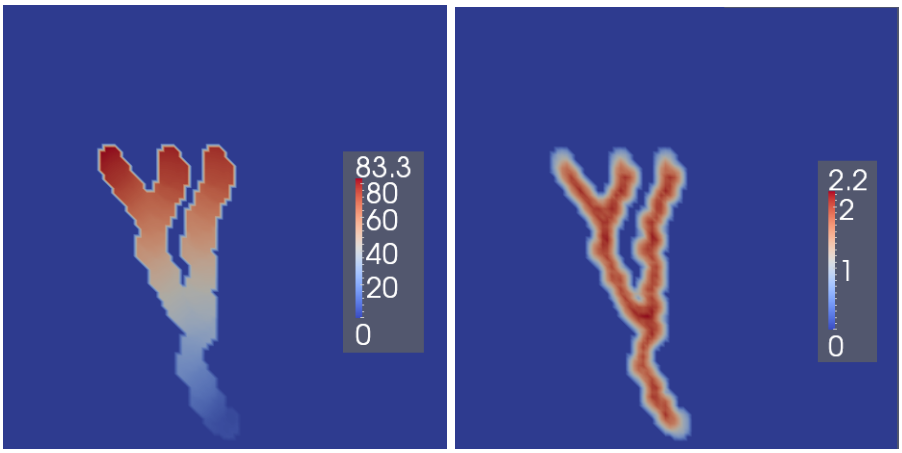

FIG. 2.2. *On the left one can see the plot of constrained distance function D in one simply connected component of the 4D segmentation, on the right one can see the plot of the constrained distance function D_B in the same simply connected component.*

move to a point in the previous time step $l-1$. The point where we moved becomes the temporary point P_T^{l-1} for time step $l-1$ and we continue the descent as above. We end the process when we cannot move from a time step N_b to a previous time step N_b-1 by decreasing value of the potential V. Then the last point of the search in the time step N_b must be some detected center $s_m^{N_b}$ and it becomes the first point of the extracted path starting in the time step N_b and ending in the time step N_e where we started the descent. As an output of this first step, we get as many extracted paths as is the number of cell identifiers in all time steps except the first one, which means that we have $\sum_{l=2}^{N_\theta} n_C^l$ steepest descent paths.

After the first step of trajectories extraction there exist many duplicated paths (representing the same cell space-time movement but for a shorter time). To illustrate the above fact, let us consider a long cell trajectory going from the first to the last 3D volume of the image sequence. Since we start the descent from centers detected at every $l = N_\theta, \ldots, 2$, and they all lay inside the branch corresponding to that cell, we obtain $N_\theta - 1$ extracted paths laying inside the same branch of the 4D spatio-temporal tree. These paths can slightly differ because the steepest descent search does not give necessarily the same set of points when starting at different time steps from different temporary points. But this does not cause any problem. Since all such paths lay in the same branch of the tree we center their points in 3D volumes by using the steepest growth direction of the constrained distance function D_B. After the centering step, the points of the shorter path become subset of points of the longer path and we can remove the shorter one just by comparing their points. The remaining path is representing the cell trajectory. After this step we obtain a set of unique cell trajectories in the sense that a mother cell representative point is presented in as many trajectories as is the number of her descendant cells.

By experiments presented in Figures 2.3-2.5 we illustrate the behavior of our cell trajectories extraction method and test its implementation.

First, we created 4D dataset (3D volumes with 100x100x100 voxels in 40 time steps) containing few cells that are moving in space and time and performing several cell divisions. The dataset contains also partial cell movements starting and ending

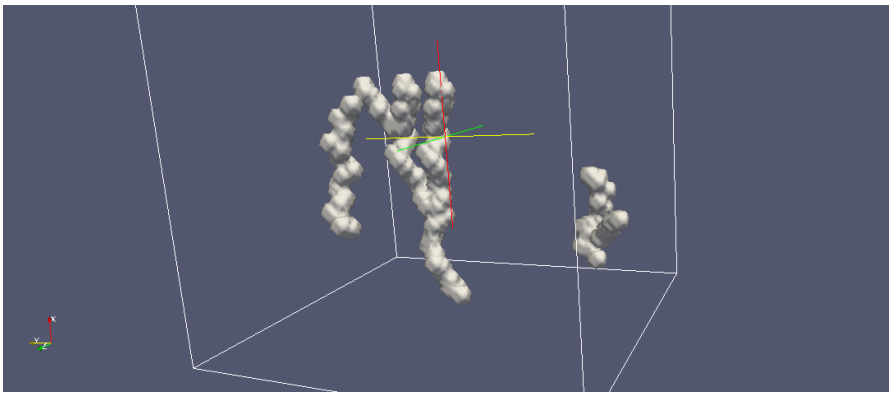

FIG. 2.3. *3D projection of the 4D segmentation of the artificial dataset where we can see cells moving and dividing inside the 4D volume.*

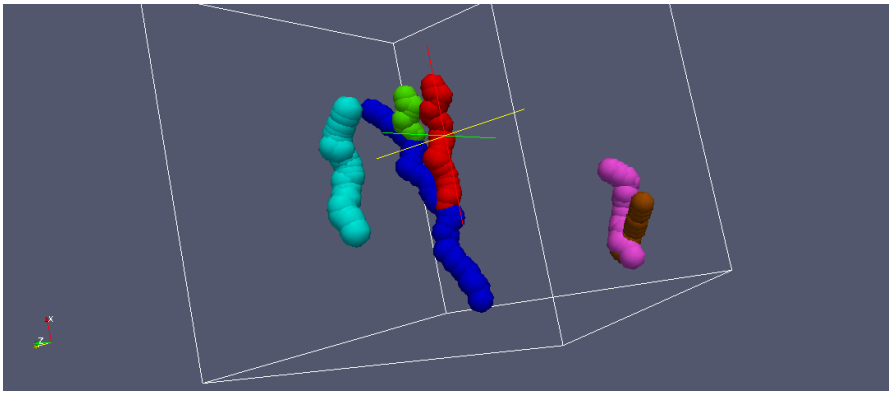

FIG. 2.4. *3D projection of the 4D points of the extracted cell trajectories.*

inside the 4D data. In fact, here we created directly the 4D segmentation, described in section 2.1, which gives us the 4D spatio-temporal tree structures corresponding to cells movements and divisions. The 3D projection of that 4D segmentation is given in Fig. 2.3. Then we performed computation of constrained distance functions D and D_B. After, we built the potential V and performed both steps (steepest descent of V and centering by D_B) of cell trajectories extraction. The resulting extracted trajectories are presented in Fig. 2.4. By different colors we plot 3D projection of 4D points (represented by small 4D spheres) of different trajectories. As we can see, all trajectories, including partial and dividing, are correctly extracted. Since the trajectories are overlapping we can see only the color of the last drawn one, e.g. the red trajectory is the same as the blue one before the first division.

In the second testing example, we created two touching branches of a 4D segmentation imitating situation that the cell nuclei are not perfectly separated due to acquisition errors. And, we tested our method in an extremal case when the branches are fully overlapped in space in two subsequent time steps (we used the same cell identifiers for those two time steps when building the 4D segmentation). In Fig. 2.5 we present the result obtained after the first step of our method, i.e. after the backtracking in time by the steepest descent direction of V. Since we perform only the

steepest descent and not the centering in this step, we can obtain two separated paths. When we center them in the second step they will remain different (although with two equal points) and thus they will represent two correctly extracted cell trajectories.

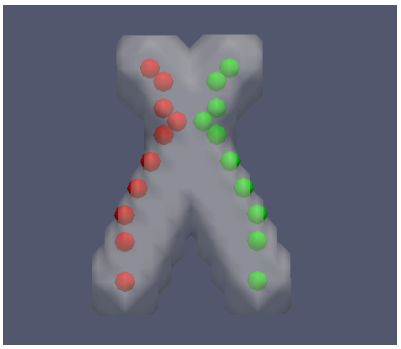

FIG. 2.5. *Separated steepest descent paths before their centering.*

3. Numerical experiment on real zebrafish embryogenesis data. For the testing of the proposed method on real data we use the representative dataset with acquisition step $d\theta = 67$ seconds, $N_\theta = 479$ number of time steps and with dimension of every 3D image 512x512x104 voxels. The real voxel size is $dx_1 = dx_2 = dx_3 = 1.37$ micrometer in every spatial direction. In the last time step $N_\theta = 479$ biologist selected manually cells forming seven presumptive organs during the zebrafish brain early embryogenesis (hypoblast, presumptive hypothalamus, ventral telencephalon, right eye, right optic stalk, left eye, left optic stalk). Using developed approach we can track those cell populations backwards (and then also forward) in time and thus follow their dynamics and clonal history.

Before tracking, all 3D images of the processed data were filtered by 10 steps of geodesic mean curvature flow (GMCF) model [8, 2, 11] and the cell nuclei identifiers were detected by 10 steps of level set center detection (LSCD) algorithm [6, 2, 11]. From several millions of cell identifiers we built the 4D segmentation and then the cell trajectories were extracted by the approach developed in section 2.

We present here Figs. 3.1 and 3.2 showing results of the tracking procedure. For the cell trajectories visualization we built software, running on graphics card, where we can fluently zoom, rotate and animate in time 3D scene even with very high number of trajectories. The trajectories are displayed as short lines (in given colors) connecting a few subsequent spatio-temporal points with a freely chosen starting time. In Fig. 3.1 we show trajectories of a subset of all cells where different colors represent different cell populations. The visualized subset represents five cell populations, presumptive hypothalamus (green), right eye (blue) , left eye (red), right optic stalk (brown) , left optic stalk (yellow). By the colored thick tubes we also plot the mean trajectory of every cell population. One can see the evolution of cells from the 0th still chaotic stage, through 100th, 200th, 300th, 400th time steps where the cells are becoming more compactly localized up to 479th time step where the cells were marked by biologist.

In Fig. 3.2 we plot another useful information which can be obtained from the cell tracking algorithm, the average velocity of the mean trajectory of every cell population. In order to compute the average velocity at time t we divide the length of trajectory (in micrometers) in time interval $[t - 10d\theta, t + 10d\theta]$ by the length of the

time interval $20d\theta$ (in seconds).

4. Conclusions. In this paper we presented new algorithm for the cell tracking in 3D+time microscopy data. We applied the tracking method to complex stages of the zebrafish early embryogenesis images. Together with [9] it is a first cell tracking method able to deal with such type of data. We discussed mathematical models forming the basis of the tracking method and present their numerical discretizations and algorithmic realizations.

Acknowledgement. We thank to all former members of EC Embryomics and BioEmergences project teams, particularly to Paul Bourgine whose continuous encouragement has helped to obtain presented results.

REFERENCES

[1] Y. Bellaïche, F. Bosveld, F. Graner, K. Mikula, M. Remešíková, M. Smíšek, *New Robust Algorithm for Tracking Cells in Videos of Drosophila Morphogenesis Based on Finding an Ideal Path in Segmented Spatio-Temporal Cellular Structures*, Proceeding of the 33rd Annual International IEEE EMBS Conference, Boston Marriott Copley Place, Boston, MA, USA, August 30 - September 3, 2011, IEEE Press, 2011

[2] P. Bourgine, R. Čunderlík, O. Drblíková, K. Mikula, N. Peyriéras, M. Remešíková, B. Rizzi, A. Sarti, *4D embryogenesis image analysis using PDE methods of image processing*, Kybernetika, Vol. 46, No. 2 (2010) pp. 226-259

[3] P. Bourgine, P. Frolkovič, K. Mikula, N. Peyriéras, M. Remešíková, *Extraction of the intercellular skeleton from 2D microscope images of early embryogenesis*, in Lecture Notes in Computer Science 5567 (Proceeding of the 2nd International Conference on Scale Space and Variational Methods in Computer Vision, Voss, Norway, June 1-5,2009), Springer (2009) pp. 38-49

[4] V. Caselles, R. Kimmel, G. Sapiro, *Geodesic active contours*, International Journal of Computer Vision 22 (1997), 67-79

[5] Y. Chen, B.C. Vemuri, L. Wang, *Image denoising and segmentation via nonlinear diffusion*, Comput. Math. Appl., 39(5-6):131149, 2000

[6] P. Frolkovič, K. Mikula, N. Peyriéras, A. Sarti, *A counting number of cells and cell segmentation using advection-diffusion equations*, Kybernetika, Vol. 43, No. 6(2007) pp. 817-829

[7] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, A. Yezzi, *Conformal curvature ows: from phase transitions to active vision*, Arch. Rational Mech. Anal. 134 (1996), 275-301

[8] Z. Krivá, K. Mikula, N. Peyriéras, B. Rizzi, A. Sarti, O. Stašová, *3D Early Embryogenesis Image Filtering by Nonlinear Partial Differential Equations*, Medical Image Analysis, Vol. 14, No. 4 (2010) pp. 510-526

[9] C.Melani, *Algoritmos de procesamiento de imagenes para la reconstruccion del desarrollo embrionario del pez cebra*, Computer Science PhD Thesis, UBA, FCEN. Buenos Aires, Argentina, 2013

[10] K. Mikula, N. Peyriéras, M. Remešíková, M. Smíšek, *4D numerical schemes for cell image segmentation and tracking*, in Finite Volumes in Complex Applications VI, Problems & Perspectives, Eds. J. Fořt et al. (Proceedings of the Sixth International Conference on Finite Volumes in Complex Applications, Prague, June 6-10, 2011), Springer Verlag, 2011, pp. 693-702

[11] K. Mikula, N. Peyriéras, M. Remešíková, O. Stašová, *Segmentation of 3D cell membrane images by PDE methods and its applications*, Computers in Biology and Medicine, Vol. 41, No. 6 (2011) pp. 326-339

[12] K. Mikula, J.Urbán, *3D curve evolution algorithm with tangential redistribution for a fully automatic finding of an ideal camera path in virtual colonoscopy*, Proceedings of the Third International Conference on Scale Space Methods and Variational Methods in Computer Vision, Ein Gedi, Israel, 2011, Lecture Notes in Computer Science Volume 6667, 2012, pp 640-652

[13] R. Mikut, T. Dickmeis, W. Driever, P. Geurts, F.A. Hamprecht, B.X. Kausler, M.J. Ledesma-Carbayo, R. Mare, K. Mikula, P. Pantazis, O. Ronneberger, A. Santos,

R. Stotzka, U. Strähle, and N. Peyriéras, *Automated Processing of Zebrafish Imaging Data: A Survey*, Zebrafish, doi:10.1089/zeb.2013.0886 (2013)

[14] E. Rouy, A. Tourin, *Viscosity solutions approach to shape-from-shading*, SIAM Journal on Numerical Analysis 29 No. 3 (1992), 867884

[15] A. Sarti, R. Malladi, J. A. Sethian, *Subjective Surfaces: A Method for Completing Missing Boundaries*, Proceedings of the National Academy of Sciences of the United States of America 12 (97) (2000), 62586263

[16] C.Zanella, M.Campana, B.Rizzi, C.Melani, G.Sanguinetti, P.Bourgine, K.Mikula, N.Peyrieras, A.Sarti, *Cells Segmentation from 3-D Confocal Images of Early Zebrafish Embryogenesis*, IEEE Transactions on Image Processing, Vol.19, No.3 (2010) pp. 770-781

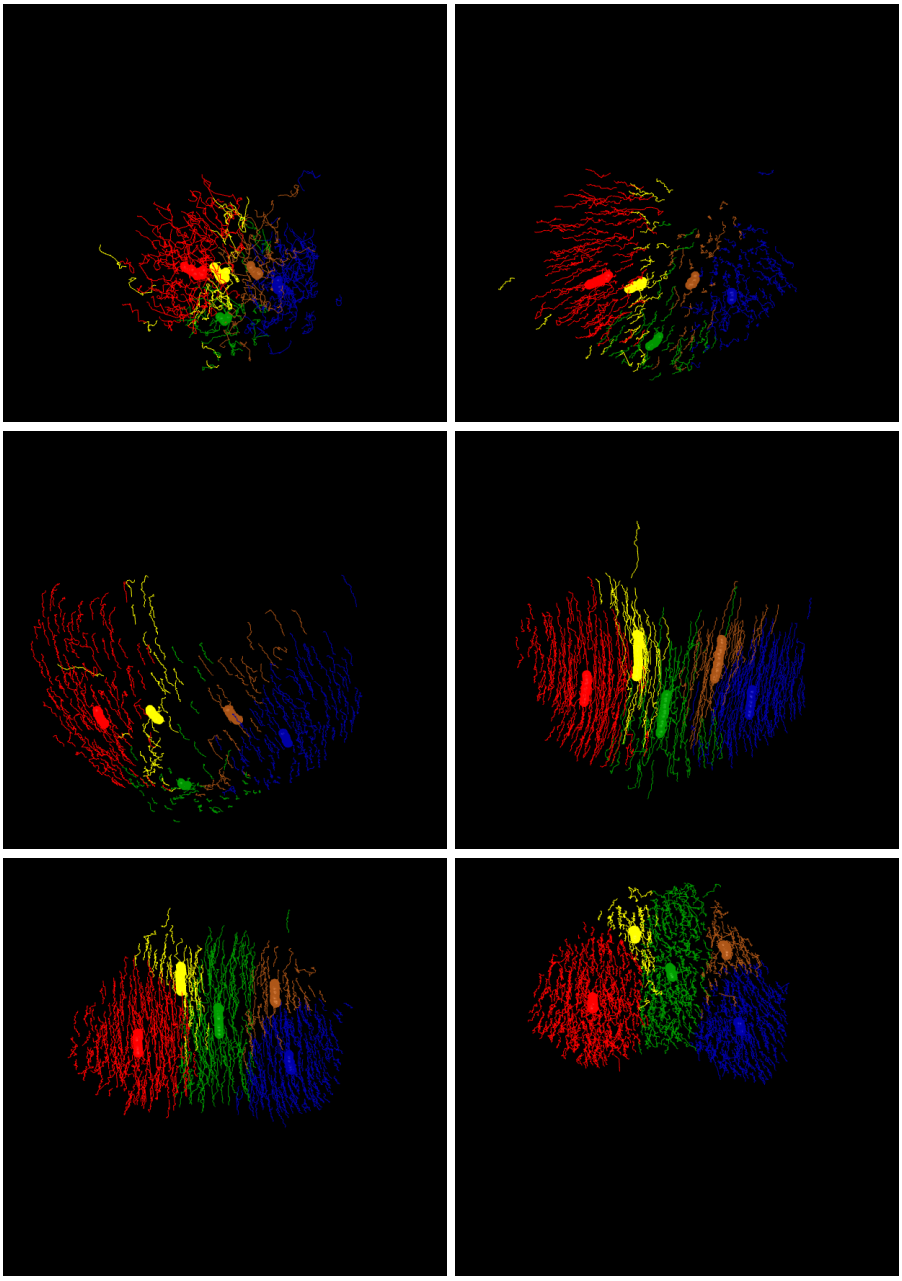

FIG. 3.1. *Visualization of the formation of presumptive organs during embryogenesis obtained by backward tracking of cells selected by biologists in time step* 479. *The visualization presents five cell populations in time steps* 0, 100, 200, 300, 400 *and* 479.

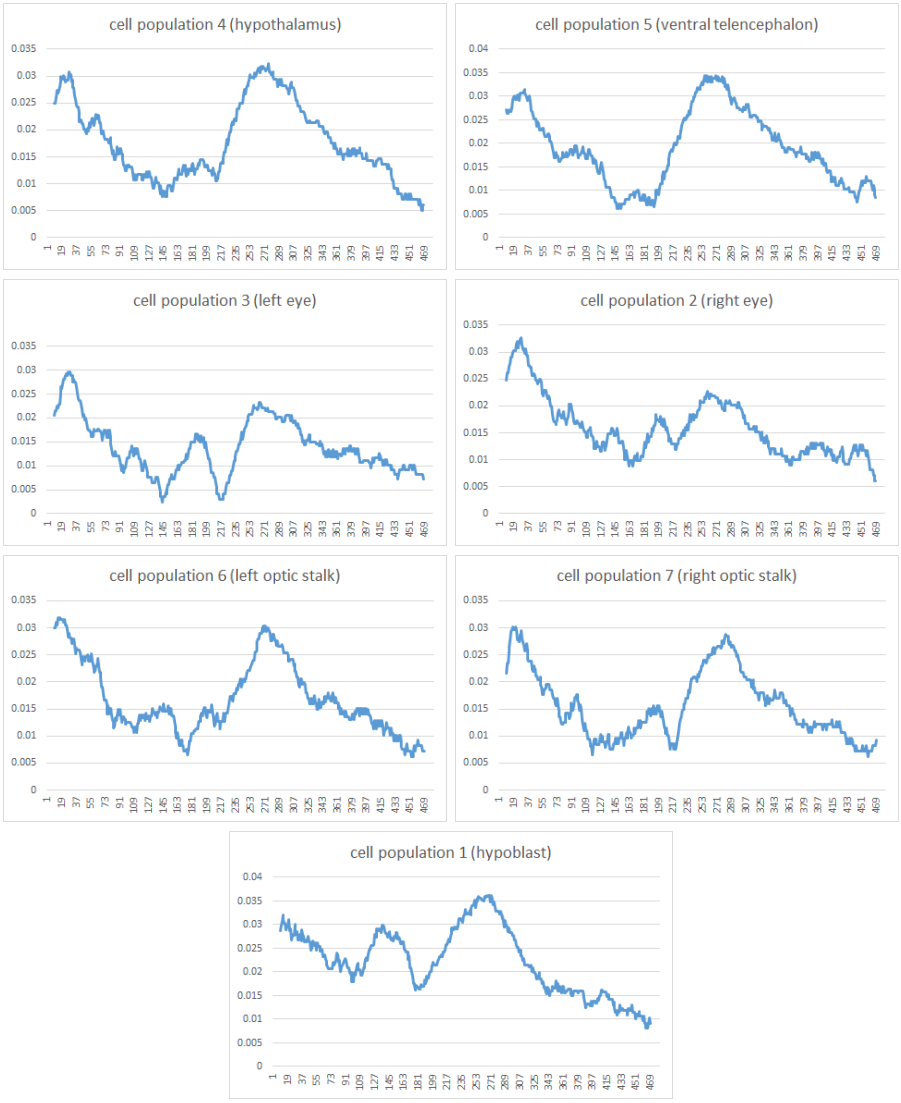

FIG. 3.2. *The graphs of the average velocity (in micrometer per second) of the mean trajectory for selected cell populations forming presumptive organs during the zebrafish brain early embryogenesis.*