BIOINFORMATICS ORIGINAL PAPER

Vol. 27 no. 4 2011, pages 564–571
doi:10.1093/bioinformatics/btq691

Data and text mining

Advance Access publication December 24, 2010

Ct3d: tracking microglia motility in 3D using a novel cosegmentation approach

Hang Xiao[1,†], Ying Li[2,†], Jiulin Du[2,*] and Axel Mosig[1,3,*]

[1]Department of Biophysics, CAS-MPG Partner Institute and Key Laboratory for Computational Biology, [2]Institute of Neuroscience, Shanghai Institutes for Biological Sciences, 320 Yue Yang Road, 200031 Shanghai, China and [3]Lehrstuhl für Biophysik, Ruhr Universität Bochum, Universitätsstraße 150, 44801 Bochum, Germany

Associate Editor: Jonathan Wren

ABSTRACT

Motivation: Cell tracking is an important method to quantitatively analyze time-lapse microscopy data. While numerous methods and tools exist for tracking cells in 2D time-lapse images, only few and very application-specific tracking tools are available for 3D time-lapse images, which is of high relevance in immunoimaging, in particular for studying the motility of microglia *in vivo*.

Results: We introduce a novel algorithm for tracking cells in 3D time-lapse microscopy data, based on computing cosegmentations between component trees representing individual time frames using the so-called *tree-assignments*. For the first time, our method allows to track microglia in three dimensional confocal time-lapse microscopy images. We also evaluate our method on synthetically generated data, demonstrating that our algorithm is robust even in the presence of different types of inhomogeneous background noise.

Availability: Our algorithm is implemented in the `ct3d` package, which is available under http://www.picb.ac.cn/patterns/Software/ct3d; supplementary videos are available from http://www.picb.ac.cn/patterns/Supplements/ct3d.

Contact: axel@picb.ac.cn; forestdu@ion.ac.cn

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on August 13, 2010; revised on December 8, 2010; accepted on December 12, 2010

1 INTRODUCTION

Capturing the motility of cells using time-lapse microscopy has become an important approach to understanding processes such as the cell cycle (Harder *et al.*, 2009), neuronal division and migration (Norden *et al.*, 2009), immune response (Cahalan and Parker, 2008) or the development of cancer (Ianzini *et al.*, 2009). Based on phase-contrast, confocal or two-photon microscopy, such *live cell imaging* protocols are now commonly established and corresponding equipment is commonly available. This has triggered the need for computational methods to quantitatively analyze time-lapse microscopy data. In this context, identifying individual cells and tracking their identities over time is one of the basic ingredients for computational analysis. Hence, *cell tracking* algorithms have attracted considerable attention in recent years (Meijering *et al.*, 2006; Miura, 2005). Here, we introduce a novel algorithm for cell tracking that allows to track cells, in particular zebrafish microglia, in 3D two-photon image sequences over time.

The majority of cell tracking algorithms, as surveyed by Meijering *et al.* (2006) or Miura (2005), deals with cell tracking in 2D over time. Methods range from linking cells identified in individual frames using different segmentation approaches to active-contour (Dufour *et al.*, 2005; Sacan *et al.*, 2008; Shen *et al.*, 2006) or level-set algorithms (Dzyubachyk *et al.*, 2008; Li *et al.*, 2008b; Mukherjee *et al.*, 2004; Nath *et al.*, 2006). The challenges imposed by the nature of the images to be analyzed lie in phenomena such as cell divisions (Al-Kofahi *et al.*, 2006; Li *et al.*, 2008a), cells entering or leaving the displayed area, or a large number of cells that needs to be tracked simultaneously. In addition, cell tracking is often complicated by background inhomogeneity, for instance due to uneven illumination (Leong *et al.*, 2003), and cells touching each other. While these issues have been addressed extensively for tracking cells in 2D, surprisingly few approaches have addressed cell tracking in 3D. Besides naive thresholding approaches, there are only few advanced approaches, such as the active-contour based method proposed by Dufour *et al.* (2005). Recently, several authors (Jaensch *et al.*, 2010; Kerekes *et al.*, 2009) proposed reliable methods for tracking centrosomes in *Caenorhabditise legans* embryos. Yet, these approaches are tailored toward tracking small, bright and circular objects which e.g. resemble a Gaussian spot of a specific size. Such assumptions, however, are not satisfied by the complex and highly variable shapes of microglia under consideration here. Cell tracking is also relevant in the context of tracking cell populations *in vitro*, which has attracted considerable attention recently (House *et al.*, 2009; Ong *et al.*, 2010; Padfield *et al.*, 2009).

The lack of methods for tracking cells in 3D has been reported as a limiting factor, for instance in the context of immunoimaging (Cahalan and Parker, 2008). Despite the well-established protocol to capture microglia, innate immune cells in the central nervous system, in 3D using two-photon microscopy following the seminal works by Nimmerjahn *et al.* (2005) and Davalos *et al.* (2005), motility analysis has been performed by (and limited to) manual estimations derived from 2D projections (Davalos *et al.*, 2008) in the numerous studies following these protocols. In fact, tracking microglia cells is complicated by several aspects. Microglia tightly contact specific brain structures in their resting state (Wake *et al.*, 2009), often making it difficult to clearly separate them from their surrounding

[*]To whom correspondence should be addressed.

[†]The authors wish it to be known that, in their opinion, the first two authors should be regarded as joint First Authors.

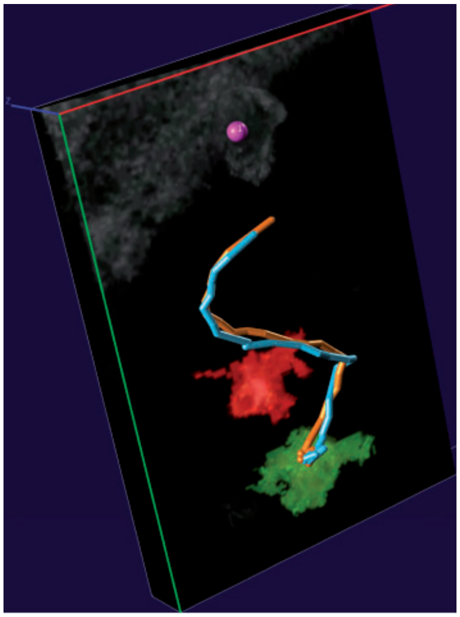


Fig. 1. 3D motion patterns of two microglia *in vivo* reconstructed using ct3d. The red and green areas indicate initial positions of the two microglia. While the red cell remains in resting state, the green cell is activated through an induced injury and migrates along a trajectory (orange line, trajectory obtained by ct3d; blue line, trajectory obtained from manual annotation) toward a site of injury (purple dot); see Supplementary Video 1 for a rendered animation of the same data.

tissue. Furthermore, the extension and retraction of the so-called microglia *processes* makes it practically impossible to separate them from other cells or surrounding tissues in a 2D projection. As we demonstrate in this study, cosegmentation-based cell tracking may overcome these difficulties and allows to reliably track microglia in 3D, both in resting state and when moving in activated state, as displayed in Figure 1.

From an algorithmic point of view, our method can be seen as a broad generalization of *thresholding methods*. Otsu's early and still commonly used approach (Otsu, 1975) picks a cut-off intensity based on the gray-value histogram of an image, considering pixel intensities below this threshold as background and pixels exceeding the threshold intensity as foreground. To deal with background inhomogeneities and objects of varying intensities, different approaches such as locally adaptive thresholding (Kim and Park, 2005) have been developed. Our approach utilizes a highly systematic way of picking local thresholds in a hierarchical representation of all possible thresholds of an image, the so-called component tree (Jones, 1999; Najman and Couprie, 2004). In order to pick local thresholds in the component tree, we compare the component trees of consecutive time frames by solving the so-called *tree assignment* problem, a natural generalization of bipartite matchings and the associated assignment problem. Comparing component trees by computing tree assignments yields a *cosegmentation* of two images; for cell tracking, cosegmentations between two time frames in a video sequence are of particular relevance.

While the term cosegmentation has been coined by Rother *et al.* (2006), our approach significantly differs from their approach, which is based on comparing histograms. On the contrary, our approach is morphological in the sense that it attempts to identify overlapping regions in two images by finding an optimal tree assignment.

Using cosegmentation has potential further applications in location proteomics beyond the cell tracking problem investigated in this article. Tree assignments as a generalization of bipartite matchings were introduced and applied by the last author recently (Mosig *et al.*, 2009), and were recently shown to be computationally hard in general (Canzar, Elbassioni and Klau, personal communication, 2010). Applying tree assignments to component trees for obtaining cosegmentations is a novel contribution in this work. In fact, cosegmentations promise to be useful in other bioimaging (and eventually image processing) applications beyond cell tracking. One straightforward application where cosegmentation is of high relevance are protein colocalization studies. Studying colocalization has recently become of relevance through the availability of corresponding two- or multi-label fluorescence microscopy (Schubert *et al.*, 2006; Zinchuk and Zinchuk, 2008) or *in situ* hybridization (Boettiger and Levine, 2009; Carson *et al.*, 2009) techniques.

We implemented our algorithm in the publicly available ct3d software package, which is accompanied by the at3d graphical user interface. In terms of applying our algorithm, this article focuses on evaluating the performance of our cosegmentation-based approach for 3D cell tracking, leaving colocalization studies as a future direction. Cell tracking performance is evaluated both on two-photon live cell imaging data displaying zebrafish microglia *in vivo*, and on synthetically generated data that allow to determine the algorithm's accuracy based on the ground truth the synthetic data were generated from.

2 METHODS

2.1 Computational methods

Our algorithm is based on representing each image $F_1,\ldots,F_N$ by its *component tree* (Jones, 1999). The component tree of an image I is obtained by considering the connected components of the thresholded versions I_θ of I under all possible thresholds θ. The set of all connected components under all thresholds is obviously hierarchically ordered by subset inclusion. This hierarchical order defines the component tree, which can be computed in linear time (Najman and Couprie, 2004). For examples of 1D images and their component trees refer to Figure 2.

Figure 3 illustrates the basic steps of our cell tracking algorithm. The outline of the algorithm is as follows: we start with computing and pruning component trees for each time frame. Then, tree assignments between each pair of consecutive component trees are computed. The tree assignments can be turned into segmentations of the original images. This produces two segmentations of each image, requiring computation of a *consensus segmentation*. The resulting unique segmentation of each image then requires a standard bipartite matching between consecutive time frames to track cell identities over time. Details of the individual steps are provided in the following paragraphs.

Computing and pruning component trees: for efficiently computing component trees, we relied on established algorithms based on a union-find data structure (Najman and Couprie, 2004). In order to reduce the size and the complexity of the component tree, we apply a *pruning* procedure to these trees. Pruning is a crucial ingredient of our algorithm, as running tree assignments on the complete component trees would be computationally too demanding. The goal of pruning is thus to eliminate as many vertices as possible, keeping only those that reflect the relevant structures of the underlying image. This is conceptually closely related to the ideas behind component filters (Salembier and Serra, 1995).

In a first pruning step, we eliminate all vertices that represent a connected component of size less than $\theta_{\min}$ or exceeding $\theta_{\max}$, where $\theta_{\min}, \theta_{\max}$ are parameters specified by the user. In a typical microscopy setting, loose upper and lower bounds on the size of the cells to be tracked are usually easy to estimate. In practice, these parameters can be chosen quite loosely, with the ratio $\theta_{\max}/\theta_{\min}$ equal to 10 or larger.

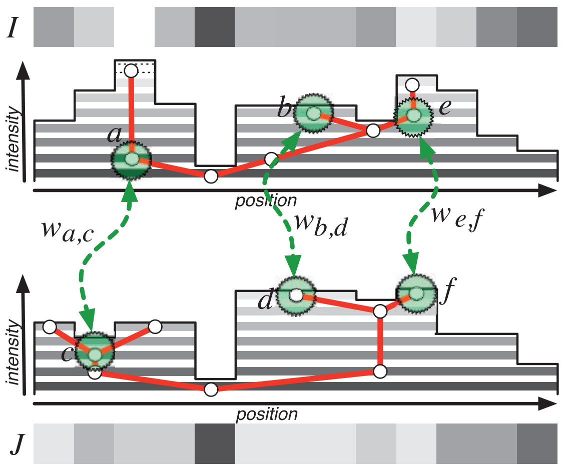


Fig. 2. Tree assignment of two (pruned) component trees for two 1D images I and J. Vertices not eliminated by the second pruning step are indicated by circles. All other non-branching vertices are eliminated in the pruning step. The tree assignment indicated by the dashed arrows is $A = \{(a,c),(b,d),(e,f)\}$ with a weight of $w_{a,c} + w_{b,d} + w_{e,f}$.

In a second pruning step, we eliminate all vertices in T_i that do not occur immediately before or immediately after a branch in the original tree (excluding leaves), as indicated in Figure 2. As this might eventually delete relevant vertices, we introduce a *single node cutoff* parameter σ, and keep every vertex v with only one child vertex w, if the symmetric difference between the areas associated with v and w comprises more than σ many pixels. As a rule of thumb, choosing $\sigma < \theta_{\min}/2$ is a good choice, which automatically limits the error rate to 50% in the worst case. For better worst-case guarantees, a smaller fraction of $\theta_{\min}$ can be chosen. Note that all results presented in this work were obtained using $\sigma = 200$.

We consider the size cutoffs $\theta_{\min}$ and $\theta_{\max}$ as parameters derived from rough estimates on the expected size of the cells to be identified. The main purpose is to eliminate vertices that result from noise in the input images, where such size cutoffs are known as *grain filters* (Vincent, 1993) as a specific type of connected operators. The expected size of noise components can be derived using random graph theory (Coupier *et al.*, 2005). As this is currently understood only for 2D images, we treat $\theta_{\min}$ as a user specified parameter.

Tree assignments: to obtain cosegmentations of the image pairs F_i and F_{i+1}, we compute a *tree-assignment* between the corresponding component trees T_i and T_{i+1}. As illustrated in Figure 2, a tree assignment A_i associates vertices (i.e. connected components) in T_i with vertices in T_{i+1}. As each assignment identifies a component in T_i and a component in T_{i+1} as a putative cell, no ancestor or descendant of the matched vertices may be part of a valid tree-assignment—otherwise, some area in an image would be occupied by two cells. If two assignments involve no overlapping components, we will also refer to them as *compatible*. Naturally, valid tree-assignments can be identified as pairwise compatible assignments. More precisely, as the quality

Algorithm `cosegmentation-track`

```
• Input: Sequence of images $F_1,\dots,F_N$; pruning parameters $\theta_{\min},\theta_{\max}$, single-node cutoff $\sigma$.
• Output: Sequence of segmented images $S'_1,\dots,S'_N$.
(1) Compute component tree $T_i$ for all $i \in [1:N]$.
(2) Prune $T_i$ to obtain $T'_i$ using $\theta_{\min},\theta_{\max},\sigma$.
(3) For each $i \in [1:N-1]$, compute $A_i = \mathrm{treeassign}(T_i, T_{i+1})$.
(4) Use $A_i$ and $A_{i+1}$ to obtain two segmentations of image $F_i$; compute consensus segmentation $S_i$ from these two.
(5) For each $i \in [1:N-1]$, compute a maximum-weighted bipartite matching between the segments in $S_i$ and $S_{i+1}$.
(6) Assign random color to each segment in $S_1$ to obtain $S'_1$. In $S'_{i+1}$, assign the same color to the segment as the one matched in $S_i$.
```

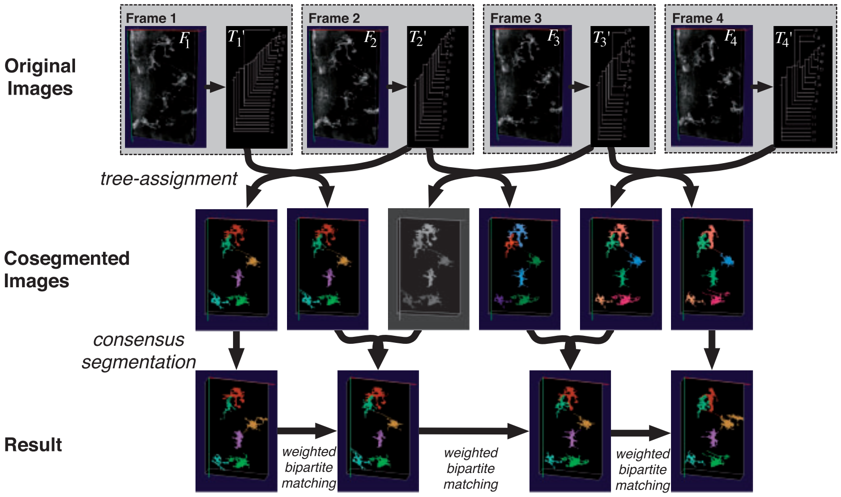


Fig. 3. Overview of complete cell tracking algorithm.

of an assignment $X_{a,b}$ can be weighted by the relative overlap $\omega_{a,b}$ of the associated components, we aim to find maximum-weighted sets of pairwise compatible assignments.

Finding maximum weighted pairwise compatible assignments naturally translates into an integer linear program by introducing a binary indicator variable $X_{a,b}$ for each possible assignment between vertices a and b. The linear program is established by introducing one constraint for each root–leaf path in each of the two trees; the sum of all variables involving any vertex along the path must be constraint to at most 1.

Weights for tree assignments: an important role is taken by the weights $\omega_{a,b}$. A straightforward choice is the 'relative overlap' between the corresponding areas $\gamma(a)$ and $\delta(b)$, i.e.

$$\omega_{a,b} := |\gamma(a) \cap \delta(b)| / |\gamma(a) \cup \delta(b)|.$$

This score is also known as the *Jaccard-Index* or the *Tanimoto Score*. An important observation is that these weights are restricted to the interval $[0,1]$. For the solution of the tree assignment, this means that in a sense also the number of segments identified will be maximized. For weighting schemes yielding unbounded values over the real numbers, large and very highly scored segments might 'overshadow' many small good assignments in the solution.

For computing all weights between any pair of vertices in two given trees, we implemented a fast algorithm utilizing the fact that 0-weighted vertex pairs do not require a variable in the linear program, along with the property that $\omega_{a,b} = 0$ implies $\omega_{a',b'} = 0$ for all descendants a' of a and b' of b. This leads to a speedup of an order of magnitude compared to a naive way of computing all pairwise weights.

Turning tree assignments into cosegmentations: assume $A = \{(a_1, b_1), \dots, (a_K, b_K)\}$ is a tree assignment between the component trees of two images I and J. Then, the areas $\gamma(a_1), \dots, \gamma(a_K)$ refer to pairwise disjoint segments in I, and can hence be considered as a segmentation of I. Correspondingly, the areas $\delta(b_1), \dots, \delta(b_K)$ induce a segmentation of J; note that the segments $\gamma(a_i)$ and $\delta(b_j)$ necessarily overlap, as they require a non-zero weight to be included in an optimal tree-assignment.

Consensus segmentation and bipartite matching: when determining a segmentation of frame i, we are confronted with two competing options; one segmentation P_i resulting from the cosegmentation of frames $i-1$ and i, the other one, Q_i, from the cosegmentation of frames i and $i+1$. We resolve this by computing a *consensus segmentation*. We generally use P_i as the starting point of a consensus segmentation. Any segment in Q_i that does not overlap with any segment P_i is supplement to P_i to obtain the consensus segmentation P'_i. This ensures that cells entering the scene in frame i can be identified in frame i (rather than frame $i+1$).

Filtering results: as for most segmentation and tracking approaches, the results obtained from the steps described above has a tendency toward *overdetection*, i.e. detecting segments that result from image noise rather than cells. To filter out those segments, we utilize *life span filtering*, i.e. we filter out all cells whose identity can be traced across less than a certain minimum number of frames. This cell filter, along with several other ways to eliminate cells with unsuitable size or volume features, follows corresponding features of the `Celltrack` software (Sacan *et al.*, 2008) for 2D cell tracking; for `ct3d`, they are implemented in the graphical user interface of the `at3d` tool shown in Figure 4.

2.2 Comparison with other approaches

While a quantitative comparison with other methods is subject of Section 3, there are noteworthy commonalities and differences with other approaches on a qualitative level as summarized below:

- *Locally adaptive thresholding:* deriving a segmentation from identifying vertices in a component tree can be seen as a way of locally adaptive thresholding; rather than determining local thresholds in pre-computed regions of interest as in Kim and Park (2005) or windows of fixed size, every component represented in the component tree is eligible to define a local threshold.

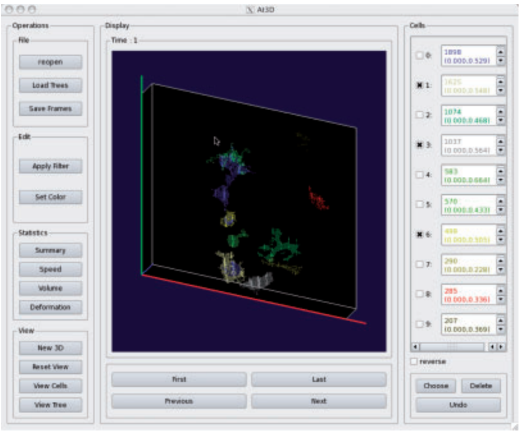


Fig. 4. Screenshot of `at3d`, which is designed for exploring cell tracking results and extracting motility parameters. It also supports features for correcting overdetection and oversegmentation. Cells can be selected and removed either individually or by filtering based on different criteria such as size or life span.

- *Absence of a global background model:* in contrast to both thresholding and level-set methods, in particular active-contour approaches, our approach does not involve any assumptions regarding the distribution of the background intensities. This is particularly useful in the presence of background inhomogeneity.
- *Alternative matching schemes:* one class of cell tracking approaches is based on computing bipartite matchings between segmentations of individual time frames. As bipartite matchings may not capture events such as cell division or cell fusion, recent works such as Padfield *et al.* (2009) introduced alternative matching schemes that are more flexible. Truly generalizing bipartite matchings, tree assignments can be seen as such alternative matching scheme. They are particularly interesting for cell tracking, as they may capture events such as cell division, cell fusion or cells entering the scene. See Supplementary Video 4 for synthetic data displaying a simplistic simulation of a cell division tracked by `ct3d`.

2.3 Implementation

We implemented component trees, tree-assignments and the complete cell tracking algorithm, in `C++` using `lp_solve`[1] for solving both the tree assignment and the weighted bipartite matching (integer) linear programs, all of which is compiled in the `ct3d` command line tool. Cell tracking results can be further explored using `at3d`, which allows the user to select and extract specific cells identified by the cell tracking procedure, and derive their motility parameters such as velocity and deformation. The `at3d` tool is implemented using the `qt` framework for graphical user interfaces. Input and output of image series is designed to be compatible with other visualization software, most notably `v3d` (Peng *et al.*, 2010) for producing rendered visualizations of the output.

2.4 Experimental materials and methods

3D time-lapse, two-photon microscopy imaging of zebrafish microglia was performed as follows: *Zebrafish preparation*. Zebrafish *Tg(ApoE:egfp)*, in which microglia express EGFP (Peri and Nüsslein-Volhard, 2008), were maintained in the National Zebrafish Resources of China (NZRC, Shanghai, China) with an automatic fish housing system (ESEN, Beijing, China) at 28°C. Embryos were raised at 28.5°C under a 14/10 h light–dark cycle in

[1] http://lpsolve.sourceforge.net/5.5/.

10% Hanks solution, which consists of (in mM) 140 NaCl, 5.4 KCl, 0.25 Na2HPO$_4$, 0.44 KH$_2$PO$_4$, 1.3 CaCl$_2$, 1.0 MgSO$_4$ and 4.2 NaHCO$_3$ (pH 7.2). They were staged as previously described (Kimmel *et al.*, 1995). Zebrafish handling procedures were approved by Institute of Neuroscience, Shanghai Institutes for Biological Sciences, Chinese Academy of Sciences.

Time-lapse imaging: for *in vivo* imaging, zebrafish larvae at 5–7 days post-fertilization (dpf) were first anesthetized with Hanks solution containing 0.02% tricaine methanesulfonate (MS222) and embedded in 1.5% low-melting point agarose. Time-lapse images of microglia were captured, via a 40× water objective mounted on a two-photon microscope with 900 nm (Prairie) or a Nikon A1R confocal microscope with 488 nm. Z-stack images, which covered the whole area of microglia in the optic tectum, were collected every 2 min at a section thickness of 1 μm. Each frame (512×512 pixels, 14 to 34 Z-stack images) was averaged four times.

3 RESULTS

We evaluated our algorithm on two types of data. First, we applied it to an *in vivo* time-lapse sequence of 3D two-photon images of zebrafish midbrain, displaying the motility of microglia; second, we applied `ct3d` to synthetically generated data for quantifying the accuracy of our cell tracking results.

Evalutation on *in vivo* data was accomplished by comparison with manually annotated trajectories of specific microglia in three datasets. Note that manual annotation is limited to trajectories, whereas boundaries of the cell volumes are almost impossible to obtain in 3D, beside systematic problems with manual annotations (Huth *et al.*, 2010). Hence, we additionally created synthetic ground truth data to further evaluate the performance of our method. We followed the procedure used in Dufour *et al.* (2005), generating (noise perturbed) elliptical objects of average intensity I_o above Poisson distributed background noise of intensity I_b. In addition to the procedure from Dufour *et al.* (2005), we created perturbed images with different types of background inhomogeneities, as shown in Figure 5: in a second set of data, we introduced a multiplicative *vignetting* effect to the data, following the vignetting model by Kang and Weiss (2000) under different focal lengths f and off-axis illumination parameters α. In a third set of data, we introduced an additive linear gradient along the x-axis of different slopes β. Each time series consists of 20 time frames, each of size $200 \times 200 \times 40$ pixels. On these data, we ran `ct3d` with size cutoffs $\theta_{\min} = 500$ and $\theta_{\max} = 5000$, and a single-node cutoff of 200 pixels. In the resulting sequences, all cells whose identity could be traced through the complete sequence were kept, while all other cells were discarded using the `at3d` tool.

3.1 Evaluation on synthetic data

Our results on the synthetically generated image sequences are summarized in Table 1 and indicate that `ct3d` is highly robust against different types and intensities of background inhomogeneities. The results suggest that `ct3d` has a tendency to identify components slightly (about 10%) larger than the actual objects, as can also be seen in the sample output in Supplementary Video 3. This is a natural consequence of pruning the component trees, where the vertex that would perfectly represent an object is unlikely to be part of the pruned tree. This effect can be reduced by smaller choice for the single-node cutoff parameter σ at the cost of higher computation time.

Running times varied between roughly 4 and 6 min, with an average of 303.42 s, for completely tracking one dataset. The majority of running time was spent on constructing the component trees and computing the overlap weights (14.23 s on average per time frame), whereas each tree assignment required less than one second on average; the pruned component trees typically comprised a few dozens of vertices.

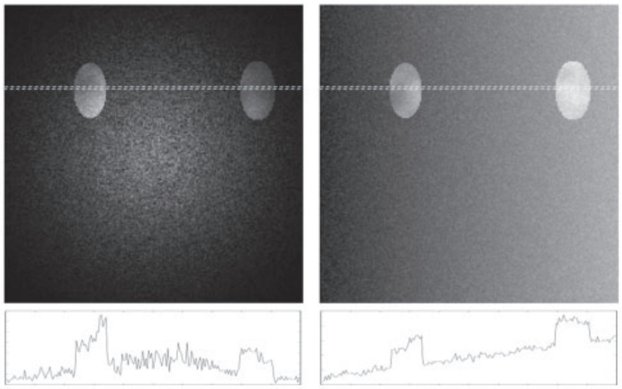


Fig. 5. Sections of synthetically generated images—Left: 2D section (top) of an image perturbed background vignetting following the Kang–Weiss model (Kang and Weiss, 2000). The dashed box indicates the position of the 1D section displayed in the lower part. Right: same setting with the background perturbed by an additive linear gradient. In both instances, `ct3d` yields reliable tracking results (see Table 1).

As a reference algorithm to compare against the performance of `ct3d`, we computed a segmentation of each time frame using the active contour approach by Chan and Vese (2001),[2] which is a well-established and state-of-the-art representative of the large family of level set methods. As shown in Table 1, this method works highly accurate in the absence of background inhomogeneity while getting less reliable with increasing levels of background inhomogeneity, as can be expected due to the involvement of a global background model.

3.2 Tracking microglia *in vivo*

Figure 6 shows a result obtained from our tracking algorithm on a time series of microglia images measured as described above. We reduced resolution by half, so that the resulting width and height varied between 146 and 250 pixels, while the depth ranged between 14 and 66 layers for each time frame; each time series comprised 30–80 time frames. Gray scale resolution was reduced from 16 bit to 8 bit. We applied `ct3d` using parameters $\theta_{\min} = 200$, $\theta_{\max} = 10\,000$ and a single-node cutoff of 200 pixels; the resulting pruned component trees contained 69 vertices on average, ranging between 40 and 168 vertices. Running times varied between roughly 2 and 10 min, with 471 s on average.

Under the given experimental protocol, the phenomenon of *overdetection*, i.e. the recognition of segments that are not microglia, is inevitable. This is due to the limited specificity of the *apoE-GFP* gene, which is also expressed in cells other than microglia in the surrounding tissue, often at comparably high levels as in microglia. Yet, `ct3d` identifies microglia as segments that can be visually distinguished from non-microglia segments by a human observer due to their characteristic shape or motion patterns. The

[2] Results were computed for parameters $\mu = 1$, $\nu = 0.7$, $\lambda_1 = 1$, $\lambda_2 = 2$, $\Delta t = 0.8$ running 100 iterations.

Table 1. Tracking results for synthetic data

	Data Set			ct3d tracking result			Chan-Vese result		
	I_o	I_b	Inhomogeneity	No. of cells	Voxel recall (%)	Error rate (%)	No. of cells	Voxel recall (%)	Error rate (%)
None	2	1	–	3.00	99.94	9.86	3.00	92.94%	7.07
	3	1	–	3.95	100.00	22.93	3.00	99.25%	0.75
	3	2	–	3.00	100.00	11.48	3.00	90.25%	9.75
	6	1	–	3.00	100.00	9.74	3.00	99.70%	0.30
	10	1	–	3.00	99.11	10.05	3.00	99.93%	0.07
Linear	3	1	$\beta=2$	3.00	100.00	10.62	2.80	80.26%	19.95%
	3	1	$\beta=3$	3.00	100.00	4.51	2.10	62.97%	39.00%
	3	2	$\beta=3$	3.00	99.97	6.83	1.95	58.27%	42.76%
	6	1	$\beta=3$	3.00	100.00	1.27	2.20	66.54%	41.26%
Vignetting	2	1	$f=100, \alpha=5e-3$	3.00	98.42	8.13	0.85	22.89%	77.25%
	2	1	$f=200, \alpha=1e-3$	3.00	99.96	9.08	1.90	56.56%	43.60%
	2	1	$f=200, \alpha=5e-3$	3.00	98.45	6.80	0.60	15.76%	84.31%
	3	1	$f=200, \alpha=1e-3$	3.00	100.00	14.25	3.00	93.70%	6.33%

To assess the quality of tracking results, we derived the number of identified cells, the *voxel recall*, i.e. what percentage of all ground truth object voxels was recovered in the tracking result, as well as the *error rate* defined as the ratio between the cardinality of the symmetric difference between tracking result and ground truth and the total number of voxels in the ground truth dataset. Left columns: each dataset displayed three cells with different intensities I_o above different levels of noise I_b, displaying different types of background inhomogeneities (see text). Middle columns: the three cells were correctly recovered under all settings by ct3d. Right columns: segmentation using the active contour approach by Chan and Vese (2001) is highly reliable in the absence of background inhomogeneity. With increasing level of inhomogeneity, voxel recall decreases, while the error rate increases. Supplementary Figure 1 displays some of the results summarized here.

graphical user interface of the at3d tool (see Fig. 4) allows to manually eliminate false positive cells from tracking results by different filtering and visual selection functions similar to those provided by Celltrack for 2D time-lapse sequences. The at3d tool also allows to manually correct for the occasionally observed events of oversegmentation, i.e. one microglia being recognized as two segments.

We used at3d to eliminate non-microglia from all six datasets and correct oversegmentation in individual frames. We were able to reconstruct trajectories of all relevant microglia that can be visually identified in the original datasets; only one dataset was affected by a sudden ‘frameshift’, i.e. the sample changing its distance to the camera in the z direction, leading to interrupted trajectories at the corresponding time frame. In few other cases, the trajectory of a microglia was interrupted in individual frames, which we could correct using at3d. Examples of the final results are given in Supplementary Videos 1 and 2.

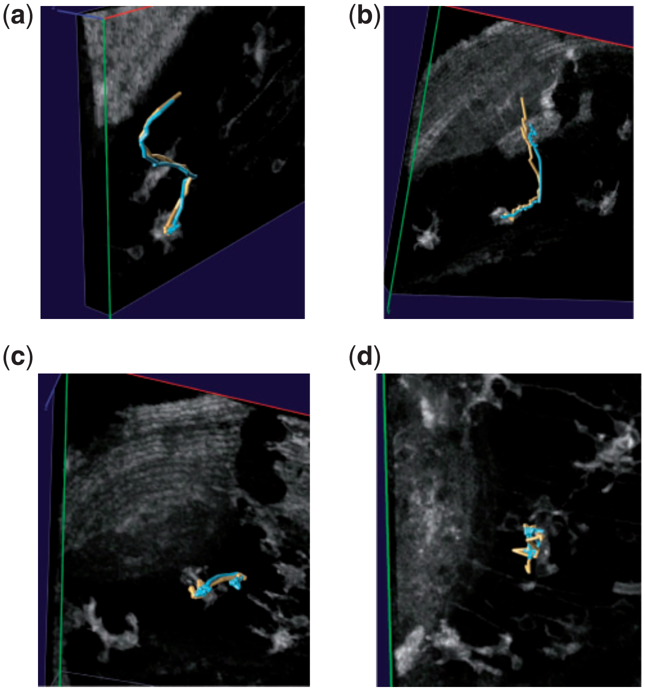
In order to quantitatively evaluate the quality of ct3d results on *in vivo* microglia data, we compared ct3d trajectories with manual annotations. To this end, we selected four microglia from four different datasets and annotated their trajectories manually using the 3D polyline markup feature of the v3d software. The root mean square distance between the ct3d and the manual trajectories turned out to be 2.9 voxels, 5.5 voxels, 2.6 voxels and 5.4 voxels, respectively, in the four different datasets. These deviations can be considered relatively small in relation to the cell diameters, which were measured as 22.8, 22, 19.6 and 44.3 voxels, respectively.

We also applied the active-contour approach from Chan and Vese (2001) to the datasets. Identifying the the four microglia in our annotated evaluation dataset required intensive tuning of the four major parameters. In fact, we could not identify a single set of parameters that works across all datasets. A parameter set tuned for a specific dataset worked comparably well on the respective dataset, but performed significantly worse or yielded no result on other datasets, see Figure 6.

4 DISCUSSION

We have presented a novel approach to tracking cells in 3D time- lapse microscopy image sequences, based on the concepts of component trees and cosegmentation. We demonstrate that this approach is robust against the numerous challenges imposed by images measured in an *in vivo* environment, and allows to identify microglia and their motion patterns in zebrafish neural tissue when combined with the at3d annotation tool. In a quantitative evaluation, we show that our approach is robust against different types of background inhomogeneities. This suggests that ct3d and at3d are potentially useful for *in vivo* imaging studies investigating other aspects than just microglia motility. In its current formulation, our cosegmentation approach relies on the assumption that the area occupied by an object overlaps between two consecutive time points. While this may not be satisfied in all cell tracking problems [e.g. when tracking centrosomes (Jaensch *et al.*, 2010)], it is a reasonable assumption for many immunoimaging-related studies.

To the best of our knowledge, our approach is the first that can identify and track microglia in live cell imaging time series. In most cases, obtaining reliable trajectories still requires manual post-processing of the output. The most notorious difficulties certainly are the complex morphology— their deformation patterns, irregular shapes and interaction with the surrounding—as well as the unspecificity of the fluorescent markers available. In this light, our approach constitutes significant progress in the sense that it has sufficient sensitivity to separate microglia form their surrounding. Yet, a fully automated approach remains a major and certainly non-trivial challenge. A first step in this direction might be the combination with level-set based approaches as utilized for 2D cell



Annotated Dataset			ct3d		Chan-Vese	
	# Frames	Cell Δ (voxel)	RMS (voxel)	# Frames	RMS (voxel)	# Frames
(a)	33	22.8	2.9	33	3.7	31
(b)	49	22	5.5	49	8.2	29
(c)	50	19.6	2.6	50	-	0
(d)	29	44.3	5.4	29	-	0

Fig. 6. Top: visualization of trajectories obtained by manual annotation (blue lines) with trajectories obtained using `ct3d` (orange lines) for microglia in activated state, see (**a**) and (**b**), as well as resting state, see (**c**) and (**d**). Bottom: quantitative comparison of trajectories obtained by `ct3d` and the active contour approach from Chan and Vese (2001). In general, `ct3d` could identify the annotated cells in all time frames (columns *# Frames*). The root mean square distance to the annotated trajectory measures a fraction of the diameter of the annotated cell (columns *cell* Δ and *RMS*), while the Chan–Vese algorithm missed varying numbers of cells or failed completely (last column). For the Chan–Vese results, a parameter set was optimized for dataset (a). This parameter set ($\mu = 1$, $\nu = .5$, $\lambda_1 = .2$, $\lambda_2 = 5$) was also applied to datasets (b) to (d). While for dataset (a), the result is comparable to `ct3d`, the annotated cell was identified in only 29 out of 49 time frames in (b). In datasets (c) and (d), the annotated cell could not be identified at all using these parameters. See Supplementary Figure S2 for further illustrations.

tracking by Nath *et al.* (2006) that might yield more accurate cell boundaries in some cases. Yet, `ct3d` promises to be a key tool for further studying open questions regarding microglia, such as to determine if and how glia and microglia share the task of finding and removing apoptotic neurons from the vertebrate brain (Peri and Nüsslein-Volhard, 2008).

Beside the direct relevance for *in vivo* time-lapse microscopy, our study indicates that our morphological approach to cosegmentation is both practical and of relevance in bioimaging. Consequentially, it appears a natural approach to apply cosegmentation to protein colocalization studies, which have attracted considerable attention in recent years following the availability of two- or multi-label fluorescence microscopy (Zinchuk and Zinchuk, 2008).

Another major experience that can be drawn from our work is the obvious potential of component trees and the closely connected theory of component filters in bioimaging. While component filters are well known to leave relevant gradients unchanged, recent work such as the results by Coupier *et al.* (2005) allow to assign a statistical significance to components observed in an image. Such concepts might be particularly useful when combining component trees with cosegmentation for judging the relevance of colocalized segments observed when comparing two component trees.

ACKNOWLEDGEMENTS

We are grateful to Dr. F. Peri for providing transgenic zebrafish line Tg(ApoE:EGFP). We acknowledge helpful remarks by Dr K. Palaniappan on implementing the level set function of the Chan-Vese method.

Funding: This work was supported by grants from the National Basic Research Program of China (2006CB806605), the Key State Research Program of China (2006CB943802), and a Wet Lab Collaboration Grant (2009GS-YJ-03) by the Shanghai Institutes for Biological Sciences.

Conflict of Interest: none declared.

REFERENCES

Al-Kofahi,O. *et al.* (2006) Automated cell lineage construction: a rapid method to analyze clonal development established with murine neural progenitor cells. *Cell Cycle*, **5**, 327–335.

Boettiger,A. and Levine,M. (2009) Synchronous and stochastic patterns of gene activation in the Drosophila embryo. *Science*, **325**, 471.

Cahalan,M. and Parker,I. (2008) Choreography of cell motility and interaction dynamics imaged by two-photon microscopy in lymphoid organs. *Annu. Rev. Immunol.*, **26**, 585–626.

Carson,J. *et al.* (2009) Automated pipeline for atlas-based annotation of gene expression patterns: application to postnatal day 7 mouse brain. *Methods*, **50**, 85–95.

Chan,T. and Vese,L. (2001) Active contours without edges. *IEEE Trans. Image Process*, **10**, 266–277.

Coupier,D. *et al.* (2005) Image denoising by statistical area thresholding. *J. Math. Imaging Vis.*, **22**, 183–197.

Davalos,D. *et al.* (2005) ATP mediates rapid microglial response to local brain injury in vivo. *Nat. Neurosci.*, **8**, 752–758.

Davalos,D. *et al.* (2008) Stable in vivo imaging of densely populated glia, axons and blood vessels in the mouse spinal cord using two-photon microscopy. *J. Neurosci. Methods*, **169**, 1–7.

Dufour,A. *et al.* (2005) Segmenting and tracking fluorescent cells in dynamic 3-D microscopy with coupled active surfaces. *IEEE Trans. Image Process*, **14**, 1396–1410.

Dzyubachyk,O. *et al.* (2008) Advanced level-set based multiple-cell segmentation and tracking in time-lapse fluorescence microscopy images. In Olivo-Marin,J.-C. *et al.* (eds) *Proceedings of IEEE International Symposium on Biomedical Imaging: From Nano to Macro IEEE*, IEEE, pp. 185–188.

Harder,N. *et al.* (2009) Automatic analysis of dividing cells in live cell movies to detect mitotic delays and correlate phenotypes in time. *Genome Res.*, **19**, 2113.

House,D. *et al.* (2009) Tracking of cell populations to understand thier spatio-temporal behavior in response to physical stimuli. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshop on Mathematical Methods in Biomedical Image Analysis*, IEEE, pp. 186–193.

Huth,J. *et al.* (2010) Significantly improved precision of cell migration analysis in time-lapse video microscopy through use of a fully automated tracking system. *BMC Cell Biol.*, **11**, 24.

Ianzini,F. *et al.* (2009) Activation of meiosis-specific genes is associated with depolyploidization of human tumor cells following radiation-induced mitotic catastrophe. *Cancer Res.*, **69**, 2296.

Jaensch,S. *et al.* (2010) Automated tracking and analysis of centrosomes in early Caenorhabditis elegans embryos. *Bioinformatics*, **26**, i13.

Jones,R. (1999) Connected filtering and segmentation using component trees. *Comput. Vis. Image Underst.*, **75**, 215–228.

Kang,S. and Weiss,R. (2000) Can we calibrate a camera using an image of a flat, textureless Lambertian surface? In *Proceeding of 6th Europe Conference Computer Vision*, Vol. 1843 of *Lecture Notes in Computer Science*, Springer, pp. 640–653.

Kerekes,R. *et al.* (2009) Automated 3-D tracking of centrosomes in sequences of confocal image stacks. In *Proceedings of Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. Vol. 1, IEEE, p. 6994.

Kim,D. and Park,J. (2005) Connectivity-based local adaptive thresholding for carotid artery segmentation using MRA images. *Image Vis. Comput.*, **23**, 1277–1287.

Kimmel,C. *et al.* (1995) Stages of embryonic development of the zebrafish. *Dev. Dyn.*, **203**, 253–310.

Leong,F. *et al.* (2003) Correction of uneven illumination (vignetting) in digital microscopy images. *J. Clin. Pathol.*, **56**, 619.

Li,K. *et al.* (2008a) Computer vision tracking of stemness. In *Proceedings of IEEE International Symposium on Biomedical Imaging*, IEEE, pp. 847–850.

Li,Y. *et al.* (2008b) Script-independent text line segmentation in freestyle handwritten documents. *Trans. Pattern Anal. Mach. Intell.*, **30**, 1313–1329.

Meijering,E. *et al.* (2006) Tracking in molecular bioimaging. *IEEE Signal Process. Mag.*, **23**, 46–53.

Miura,K. (2005) Tracking movement in cell biology. *Adv. Biochem. Eng. Biotechnol.*, **95**, 267–295.

Mosig,A. *et al.* (2009) Tracking cells in live cell imaging videos using topological alignments. *Algorithms Mol. Biol.*, **4**, 1–9.

Mukherjee,D. *et al.* (2004) Level set analysis for leukocyte detection and tracking. *IEEE Trans. Med. Imaging*, **13**, 562–572.

Najman,L. and Couprie,M. (2004) Quasi-linear algorithm for the component tree. *SPIE Vis. Geom. XII*, **5300**, 98–107.

Nath,S.K. *et al.* (2006) Robust tracking of migrating cells using four-color level set segmentation. In *Advanced Concepts for Intelligent Vision Systems*, Vol. 4179 of *Lecture Notes in Computer Science*, Springer, pp. 920–932.

Nimmerjahn,A. *et al.* (2005) Resting microglial cells are highly dynamic surveillants of brain parenchyma in vivo. *Science*, **308**, 1314.

Norden,C. *et al.* (2009) Actomyosin is the main driver of interkinetic nuclear migration in the retina. *Cell*, **138**, 1195–1208.

Ong,L. *et al.* (2010) Tracking of cell population from time lapse and end point confocal microscopy images with multiple hypothesis Kalman smoothing filters. In *Proceedings of Computer Vision and Pattern Recognition Workshops (CVPRW)*, IEEE, pp. 71–78.

Otsu, N. (1975). A threshold selection method from gray-level histograms. *Automatica*, **11**, 285–296.

Padfield,D. *et al.* (2009) Coupled minimum-cost flow cell tracking. In *Proceedings of Information Processing in Medical Imaging*, Vol. 5636 of *Lecture Notes in Computer Science*, Springer, p. 374.

Peng,H. *et al.* (2010) V3D enables real-time 3D visualization and quantitative analysis of large-scale biological image data sets. *Nat. Biotechnol.*, **28**, 348–353.

Peri,F. and Nüsslein-Volhard,C. (2008) Live imaging of neuronal degradation by microglia reveals a role for v0-ATPase a1 in phagosomal fusion in vivo. *Cell*, **133**, 916–927.

Rother,C. *et al.* (2006) Cosegmentation of image pairs by histogram matching - incorporating a global constraint into mrfs. In *Proceedings Conference on Computer Vision and Pattern Recognition*. Vol. 1, IEEE, pp. 993–1000.

Sacan,A. *et al.* (2008) Celltrack: an open-source software for cell tracking and motility analysis. *Bioinformatics*, **24**, 1647–1649.

Salembier,P. and Serra,J. (1995) Flat zones filtering, connected operators, and filters by reconstruction. *IEEE Trans. Image Process.*, **4**, 1153–1160.

Schubert,W. *et al.* (2006) Analyzing proteome topology and function by automated multidimensional fluorescence microscopy. *Nat. Biotechnol.*, **24**, 1270–1278.

Shen,H. *et al.* (2006) Automatic tracking of biological cells and compartments using particle filters and active contours. *Chemometrics Intell. Lab. Syst.*, **82**, 276–282.

Vincent,L. (1993) Grayscale area openings and closings, their efficient implementation and applications. In *Proceedings of EURASIP Workshop on Mathematical Morphology and its Applications to Signal Processing*, pp. 22–27.

Wake,H. *et al.* (2009) Resting microglia directly monitor the functional state of synapses in vivo and determine the fate of ischemic terminals. *J. Neurosci.*, **29**, 3974.

Zinchuk,V. and Zinchuk,O. (2008) Quantitative colocalization analysis of confocal fluorescence microscopy images. *Curr. Protoc. Cell Biol.*, **39**, 4–19.