

A Method for Actin Filament Tracking in Fluorescent Microscopy Images^{*}

Danil A. Kononykhin¹, Valentina Yu. Berg², Denis A. Ovsyannikov²,
Andrey S. Krylov¹, and Dmitry V. Sorokin¹

¹ Laboratory of Mathematical Methods of Image Processing, Faculty of Computational
Mathematics and Cybernetics,

Lomonosov Moscow State University, Moscow, Russia

² Institute of Immunology and Physiology, Ural Branch of the Russian Academy of Sciences,
Yekaterinburg, Russia

danil.kononyhin@mail.ru, valeoupi@gmail.com,

d.ovsjannikov@gmail.com,

kryl@cs.msu.ru, dsorokin@cs.msu.ru

Abstract. The automated tracking of subcellular structures in live microscopy image sequences is an actual problem in many biological research areas. A universal solution for this problem still does not exist due to a huge variety of data of different nature. In this work, we propose an algorithm for tracking actin filaments in 2D fluorescent image sequences. The filaments are moving in a random and abrupt manner frequently crossing each other. We used steerable filters based ridge detection followed by crossing filaments correction algorithm for filaments detection. The tracking was performed using a greedy nearest neighbor method. The quantitative evaluation of our approach was performed on several manually annotated image sequences using object tracking quality metric MOTA. It was shown that the proposed approach outperforms an existing approach in tracking accuracy. In addition, the proposed approach allows processing crossed filaments, unlike the existing methods.

Keywords: Actin Filaments, Steerable Filters, Tracking, Fluorescent Microscopy.

1 Introduction

Research on muscle functions is conducted for many years [5], yet there are still incurable muscle diseases. In 1954 sliding filament theory [9] was proposed, which explained the inner work of muscles. Two proteins actin and myosin are responsible for muscle contraction. To analyze their interaction *in vitro motility assay* (IVMA) is used. Myosin is immobilized on a glass while fluorescent-stained actin filaments are

Copyright © 2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

* The computer algorithms for biomedical image processing and analysis in this work were funded by RFBR, CNPq and MOST according to the research project No. 19-57-80014 (BRICS2019-394), the filaments experimental data preparation and acquisition was funded by RFBR project 18-015-00252, and State Program No. AAAA-A19-119070190064-4.

immersed in a solution with ATP (Adenosine Triphosphate). Actin filaments start moving when they interact with myosin in the presence of ATP. By analyzing the velocity of actin filaments different mutations of myosin and actin can be studied. To analyze the velocities filaments must be tracked. An example of images with filaments are given in Fig. 1.

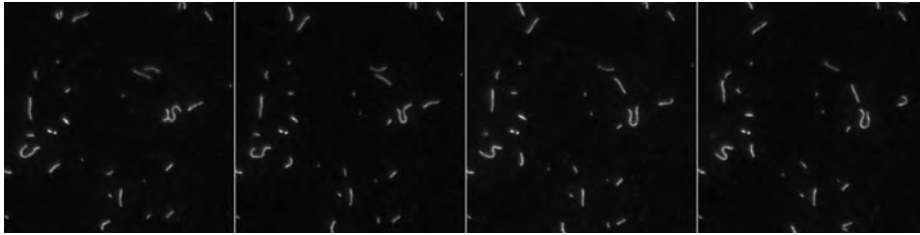


Fig. 1. First four frames from a sequence.

Most existing methods [13] solve the tracking task in two stages: first, the objects are detected in the image sequence, second, the tracks are formed by the same objects from several consequent frames using data association techniques. The most accurate technique that gives the optimal solution is multiple-hypothesis tracking (MHT), yet it has polynomial complexity, and thus computationally intensive. That's why a lot of methods were designed to approximate this optimal solution, such as multi-dimensional assignment problem (MAP) solver [7], noniterative greedy assignment (NGA) [12], two-step linear assignment procedure (LAP) [11], greedy nearest-neighbor (GNN) linking [3].

In [1], the authors proposed an object tracking method named FAST (Fast Automated Spud Tracker) that was specifically designed to solve the actin filament tracking problem. The FAST method relies on the generalized nearest neighbor method for data association and uses additional information about filaments (their shape and behavior) to make data association step more robust. The filament detection stage is based on background subtraction and skeletonization. The method showed a good performance; however, does not track intersecting filaments and makes a lot of false negatives when detecting filaments.

In this work, we propose a new method for actin filament tracking. We use steerable filters with Canny-like criteria [10] to detect filaments. In addition, we address the problem of crossed filaments tracking by aggregating information from several consecutive frames. The data association step is performed with the greedy nearest neighbor approach using distance between filament centroids. Our approach has been quantitatively evaluated using manually annotated real microscopy image data and a comparison with the existing approach in [1] has been performed.

2 Methods

Our filament tracking approach consists of three stages: filament detection, correction of crossed filaments, and data association.

2.1 Filament Detection

We use steerable filters with Canny-like criteria [6, 10] to detect the filaments in each image of the sequence. The algorithm consists of the following steps: ridge detection, non-maximum suppression, double thresholding, and hysteresis.

For ridge detection we use steerable filters [8]. To detect a particular feature on an image we need to convolve the image with a filter representing the ideal template of this feature rotated at different angles. Thus, a high magnitude of the resulting convolution with the rotated template indicates the presence of the feature and its orientation:

$$\theta^*(\mathbf{x}) = \arg \max_{\theta} (f(\mathbf{x}) * h(\mathbf{R}_{\theta}\mathbf{x})), \quad (1)$$

$$r^*(\mathbf{x}) = f(\mathbf{x}) * h(\mathbf{R}_{\theta^*}\mathbf{x}), \quad (2)$$

where $f(\mathbf{x})$ is the initial image, $h(\mathbf{x})$ is the filter, r^* is the filter response, θ^* is the feature orientation, and \mathbf{R}_{θ} is the rotation matrix.

To detect the feature in all orientations one needs to have a very fine angle grid resulting a number of convolutions with templates oriented at various angles. The steerable filters are used to simplify the computations as they can be rotated efficiently by taking a linear combination of a small number of filters.

In this work, the following filter was chosen [10]:

$$h(x, y) = -\sqrt{\frac{3}{4\pi}}\sigma \left(g_{yy} - \frac{g_{xx}}{3} \right) = \alpha \cdot g_{xx} + \beta \cdot g_{yy}, \quad (3)$$

where $g(x, y)$ is the Gaussian function, $\alpha = \frac{1}{3}\sqrt{\frac{3}{4\pi}}\sigma$, $\beta = -\sqrt{\frac{3}{4\pi}}\sigma$. Visualization of the $h(x, y)$ filter is shown in Fig. 2.

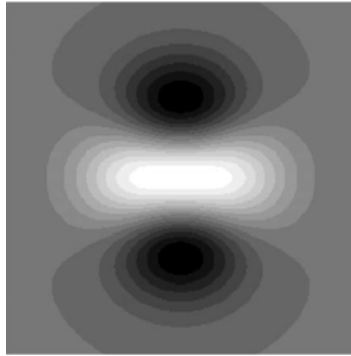


Fig. 2. The steerable filter $h(x, y)$ used for ridge detection.

We can derive the equation to compute this filter at any orientation by applying Fourier transform to a rotated filter, combining like terms, and applying inverse Fourier

transform:

$$\begin{aligned}
 h(\mathbf{R}_\theta \mathbf{x}) = & \cos^2 \theta \cdot \underbrace{(\alpha \cdot g_{xx} + \beta \cdot g_{yy})}_{q_1(\mathbf{x})} + \\
 & + \cos \theta \sin \theta \cdot \underbrace{(2 \alpha \cdot g_{xy} - 2 \beta \cdot g_{xy})}_{q_2(\mathbf{x})} + \\
 & + \sin^2 \theta \cdot \underbrace{(\alpha \cdot g_{yy} + \beta \cdot g_{xx})}_{q_3(\mathbf{x})}.
 \end{aligned} \tag{4}$$

Thus, plugging Eq. (4) into Eq. (1) we can analytically maximize the filter response by θ at every pixel of the image after computing only three convolutions with the filters $q_1(\mathbf{x})$, $q_2(\mathbf{x})$, and $q_3(\mathbf{x})$.

After getting filter response in every pixel of the image, we thin the found ridges by non-maximum suppression (we suppress all pixel values that have lower intensity than their neighbors along the θ angle). Then, we apply double thresholding and hysteresis to remove the noisy detections. The low threshold is computed using the Otsu method, while the high threshold is a parameter of the algorithm and is typically set in a way that the thresholded image has 90% of the brightest pixels from the image after non-maximum suppression step.

As the result of the hysteresis we get the detected filaments. They can be seen in Fig. 3 as magenta lines.

2.2 Correction of Crossed Filaments

In case the filaments cross, the detection approach described above causes splitting of one of the filaments or merging two neighboring filaments (see Fig. 3). In the third frame of Fig. 3 the filaments splitting problem is shown. In the second and the fifth frames of Fig. 3 one can see merged filaments.

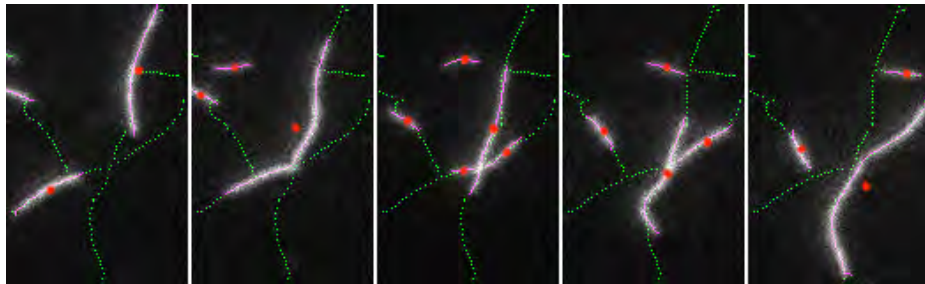
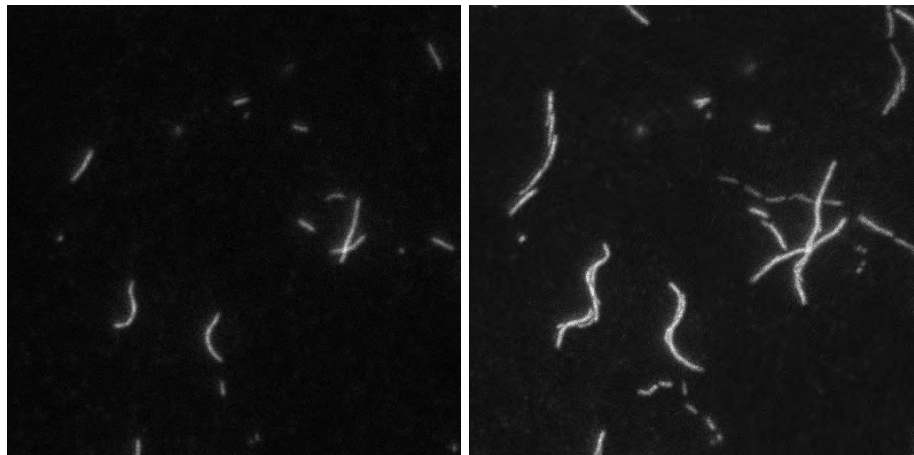
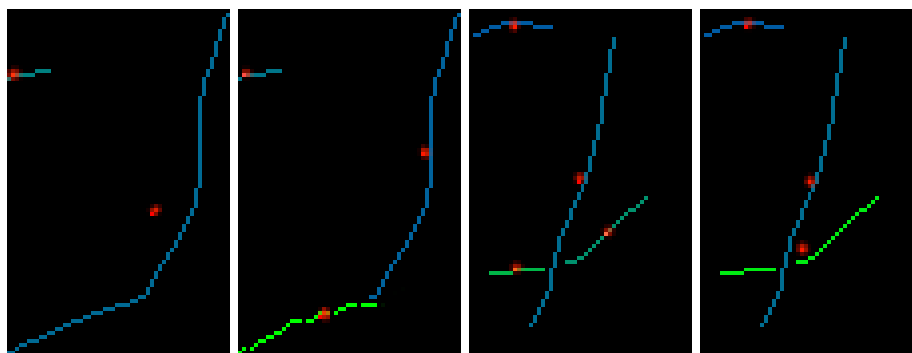


Fig. 3. Five frames are visualized. Original input images are overlaid by detected filaments and found traces. Red dots mark filament centroids. Magenta lines are detected filaments. Green lines are filament traces.



(a) The 11-th frame of the image sequence. (b) The maximum projection of 5 frames (from 10-th to 15-th).

Fig. 4. The illustration of the maximum projection image from five consecutive frames. The maximum projection image is used to construct filament traces.



(a) False merging filaments problem. Two filaments that were merged during the detection stage are on the left. The result of the false merging correction algorithm where the filament is split into two is on the right. (b) False splitting filaments problem. On the left, the horizontal filament is split into two filaments. The result of false splitting correction algorithm where two filaments are merged into one is shown on the right.

Fig. 5. False merging filaments problem (a), and false splitting filaments problem (b). The detected filaments are shown with different colors. Red dots mark the centroids of the filaments.

To address this problem, we use the information from several consecutive frames (the number of frames is the parameter of the algorithm) by taking the maximum intensity over time in each pixel (maximum projection). An example of the maximum projection can be seen in Fig. 5.

The resulting maximum projection images represent the filament motion traces within the specified period. To analyze these structures we use FIRE algorithm [14]. This algorithm was created to join ridge-like structures of similar direction that separated on the ridge detection stage.

First, the maximum projection image is binarized. We consequentially apply the steerable (see Section 2.1) and median (5×5) filters to the image. After that, the image is thresholded by the global Otsu threshold. Then, the distance transform is applied followed by Gaussian smoothing ($\sigma = 1.5$) resulting in a smoothed distance transform E . After that, the nucleation points are found as the local maxima in a square neighborhood of the radius s_{max} (which is set to 5 in our method). Then we form a set of branches for every nucleation point. The branch begins in the nucleation point and follows the directions given by local maxima of the distance transform E . The initial direction is given by the local maxima on a border of a square neighborhood of the nucleation point. The branch extension process ends when the branch meets the background or another nucleation point. For the formal description of this approach we refer to [2]. Finally, the co-directional branches are merged and branches shorter than 5 pixels are removed resulting in filament traces (see Fig. 3).

After the filament traces are found each filament in every frame is assigned to its closest trace. Thus, one filament trace can have a lot of filaments associated with it. Then each trace can be described by frame numbers where it appeared t_i and the number of filaments assigned to the trace in each frame n_{t_i} :

$$\{t_1 : n_{t_1}, t_2 : n_{t_2}, \dots, t_k : n_{t_k}\}, \quad (5)$$

where k is the number of consecutive frames that are taken for maximum projection (we typically use $k = 5$).

For example, $\{11: 1, 12: 2, 13: 1, 14: 1, 15: 1\}$ means that in the 11-th, 13-th, 14-th, and 15-th frames the trace was assigned to one filament, and in the 12-th frame it was assigned to two filaments.

Ideally, each trace in every frame must have exactly one filament that forms the trace. However, we can distinguish two kinds of errors: zero filaments associated with the trace in a frame ($n_{t_i} = 0$), and more than one filaments associated with the trace in a frame ($n_{t_i} > 1$). When we have zero filaments in a frame, it means that the filament which was supposed to be detected and assigned to this trace was not detected or merged with some other filament. We call this false merging problem (see Fig. 5a). In case we have more than one filament associated with the trace in one frame, it means that the filament was incorrectly split into two separate filaments (see Fig. 5b). The solution for both problems is described below. To simplify further descriptions, let us define m as the mean length of all filaments from the current trace where $n_{t_i} = 1$.

False Merging Filaments Correction. First, we find all traces with $n_{t_i} = 0$ at least for one frame number. We restore lost filaments using the following procedure. If $n_{t_{i-1}} = n_{t_{i+1}} = 1$, we can construct filament along the trace with length m and centroid computed as mean of centroids of filaments in the neighboring frames t_{i-1} and t_{i+1} . If the missing filament is in the first or last frame ($n_{t_1} = 0$ or $n_{t_k} = 0$), we form the filament by cutting a piece of the trace of the length m from its start (or end). In case

the miss of the filament was caused by merging with another filament, we have to correct the length of the latter one by removing the part corresponding to the reconstructed filament. Thus, we get two distinct filaments from one merged filament (see Fig. 5a).

False Splitting Filaments Correction. We find all traces with $n_{t_i} > 1$ at least for one frame number. For each trace, for every frame t_i where $n_{t_i} > 1$ we calculate the sum of filaments lengths in this frame l_{t_i} . If the length l_{t_i} is close to the mean length m (their difference is less than 20 pixels), we merge these filaments into a single one (see Fig. 5b).

Algorithm Summary. The described correction steps are used iteratively using the following algorithm:

1. Construct the set of filament traces from one aggregated image.
2. Assign all filaments to the traces.
3. Apply false splitting filaments correction to all traces.
4. Apply false merging filaments correction to all traces.
5. If any filament was split or merged, mark the corresponding traces as changed, and mark them unchanged otherwise.
6. If all traces are unchanged, then the algorithm is finished. Otherwise, return to step 3.

2.3 Filament Association

For filament association we use the generalized nearest neighbor algorithm. The algorithm is applied iteratively from the first frame to the last frame.

Suppose that all the tracks are constructed till t -th frame $\{\gamma_1, \dots, \gamma_N\}$. We need to continue tracks from t frame to $t + 1$ frame. Let us describe the procedure for the track γ_i and its last filament f .

First, we compute distances between the centroids of the filament f and all the filaments from the frame $t + 1$. Then, we filter out all the filaments which length differ more than 1.5 times from the length of the last filament in this track. We do not filter out the filaments shorter than 25 pixels, as the length of the filaments can vary up to 10 pixels between the frames due to the noise. After that, we filter out all filaments that are too far. We do not consider the filaments that are further than the search radius of $3r$, where r is the mean of all filament displacements in the track γ_i . Finally, for the filament f we choose the closest filament from the remaining candidates from the frame $t + 1$. If there is no such filament then the track is ended. The procedure is repeated greedily for all the tracks γ_i . All filaments from frame $t + 1$ that were not assigned to any track are considered as the beginnings of the new tracks.

3 Results

3.1 Dataset

The data was kindly provided by the Institute of Immunology and Physiology, Ural Branch of the Russian Academy of Sciences. The image sequences consist of 30 frames

with the size of 512×512 . The time interval between the frames is 300 ms. For the algorithm evaluation, three sequences with 123 tracks were manually annotated.

3.2 Evaluation Metric

We used the MOTA metric from the CLEAR MOT metrics [4] for quantitative evaluation of the tracking results:

The multiple object tracking accuracy (MOTA) is defined as

$$\text{MOTA} = 1 - \frac{\sum_t m_t}{\sum_t g_t} - \frac{\sum_t \text{fp}_t}{\sum_t g_t} - \frac{\sum_t \text{mme}_t}{\sum_t g_t} \quad (6)$$

Here, m_t is the number of misses, fp_t is the number of false positives, mme_t is the number of mismatch errors, g_t is the number of ground truth objects in frame t .

Note, that MOTA can be negative (because there could be more false positives and misses than ground truth objects).

3.3 Experiments

We calculated the values of MOTA metric on three manually annotated image sequences for the FAST method [1], and the proposed method.

In Table. 1 the results of MOTA are presented. Our method has higher MOTA values. It is important to note that the proposed approach is able to work with crossed filaments, while FAST method does not consider them at all, which partly affects such a difference in MOTA values.

In Fig. 6 the tracking results for the first 24 frames of one of the sequences are visualized. To make images less cluttered the finished tracks are not shown.

Table 1. MOTA metrics computed for the FAST method [1] and the proposed methods.

Method	Seq. 1	Seq. 2	Seq. 3
FAST	-0.316	-0.177	-0.169
Proposed method	0.422	0.482	0.589

4 Conclusion

In this paper, we have presented a method for tracking actin filaments in fluorescence microscopy image sequences. The filament detection is based on ridge detection using steerable filters with Canny-like criteria. In addition, the filament detection includes crossed filaments correction algorithm based on filament trace analysis. The tracking is based on the greedy nearest neighbor approach that takes into account the distance between filaments and their lengths. We compared the results of the proposed method with the existing approach [1] using MOTA metric on the manually annotated data. The quantitative evaluation showed that our approach outperforms the existing approach for our data in tracking accuracy.

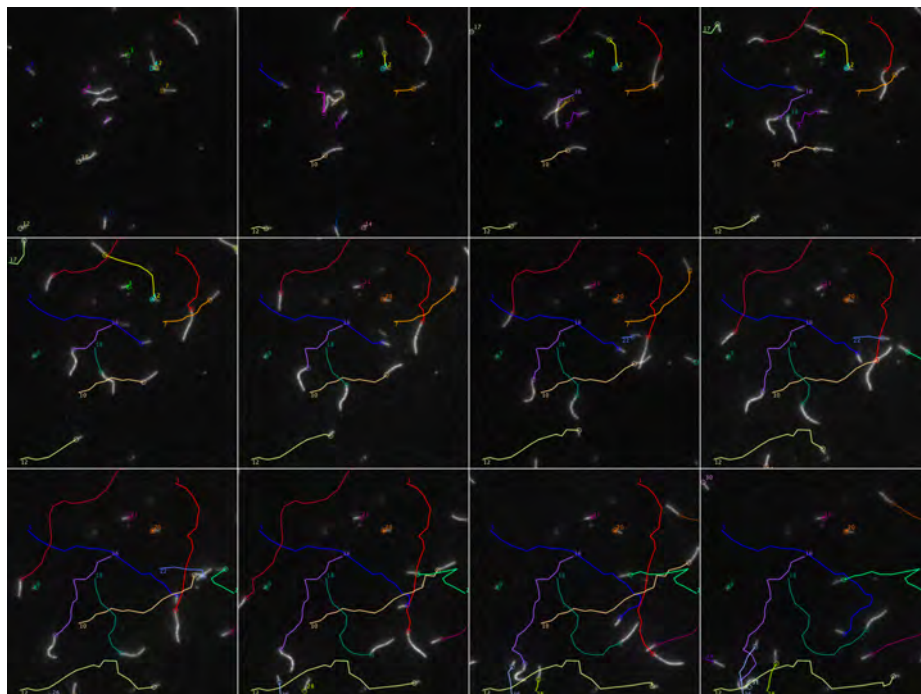


Fig. 6. Filaments tracking result. From left to right from top to bottom every second frame is shown.

References

1. Aksel, T.e.a.: Ensemble force changes that result from human cardiac myosin mutations and a small-molecule effector. *Cell Reports* **11**, 910–920 (2015)
2. Anguiano, M., Castilla, C., Maška, M., Ederra, C., Fernández-Marqués, J., Peláez, R., Rouzaut, A., Muñoz-Barrutia, A., Kozubek, M., Ortiz-de-Solórzano, C.: Characterization of the role of collagen network structure and composition in cancer cell migration. In: 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). pp. 8139–8142 (2015)
3. Apgar, J., Tseng, Y., Fedorov, E., Herwig, M.B., Almo, S.C., Wirtz, D.: Multiple-particle tracking measurements of heterogeneities in solutions of actin filaments and actin bundles. *Biophysical Journal* **79**(2), 1095 – 1106 (2000)
4. Bernardin, K., Stiefelhagen, R.: Evaluating multiple object tracking performance: The clear mot metrics. *EURASIP Journal on Image and Video Processing* **2008**, 1–10 (2008)
5. Bopp, N., Scheid, L.M., Fink, R., Rohr, K.: Determination of the maximum velocity of filaments in the in vitro motility assay. *Frontiers in physiology* **10**, 279 (2019)
6. Canny, J.F.: A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-8**, 679–698 (1986)
7. Feng, L., Xu, Y., Yang, Y., Zheng, X.: Multiple dense particle tracking in fluorescence microscopy images based on multidimensional assignment. *Journal of Structural Biology* **173**(2), 219 – 228 (2011)

10 D.A. Kononykhin et al.

8. Freeman, W.T., Adelson, E.H.: The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13**(9), 891–906 (1991)
9. Huxley, A.F., Niedergerke, R.: Structural changes in muscle during contraction: Interference microscopy of living muscle fibres. *Nature* p. 971–973 (1954)
10. Jacob, M., Unser, M.: Design of steerable filters for feature detection using canny-like criteria. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**(8), 1007–1019 (2004)
11. Jaqaman, K., Loerke, D., Mettlen, M., Kuwata, H., Grinstein, S., Schmid, S., Danuser, G.: Robust single-particle tracking in live-cell time-lapse sequences. *Nature Methods* **5**, 695–702 (2008)
12. Shafique, K., Shah, M.: A noniterative greedy algorithm for multiframe point correspondence. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27**(1), 51–65 (2005)
13. Smal, I., Meijering, E.: Quantitative comparison of multiframe data association techniques for particle tracking in time-lapse fluorescence microscopy. *Med Image Analysis* pp. 163–189 (2015)
14. Stein, A., Vader, D., Jawerth, L., Weitz, D., Sander, L.: An algorithm for extracting the network geometry of 3d collagen gels. *Journal of Microscopy* **232**, 463–75 (2009)