Tracking Multiple Neurons on Worm Images

Toufiq Parag[a], Victoria Butler[b] and Dmirti Chklovskii[a]

[a]Janelia Farm Research Campus-HHMI, Ashburn, VA, USA;
[b]Medical Research Council Laboratory of Molecular Biology, University of Cambridge, UK.

ABSTRACT

We are interested in establishing the correspondence between neuron activity and body curvature during various movements of *C. Elegans* worms. Given long sequences of images, specifically recorded to glow when the neuron is active, it is required to track all identifiable neurons in each frame. The characteristics of the neuron data, e.g., the uninformative nature of neuron appearance and the sequential ordering of neurons, renders standard single and multi-object tracking methods either ineffective or unnecessary for our task. In this paper, we propose a multi-target tracking algorithm that correctly assigns each neuron to one of several candidate locations in the next frame preserving shape constraint. The results demonstrate how the proposed method can robustly track more neurons than several existing methods in long sequences of images.

1. INTRODUCTION

Movement is the main output of the *C. Elegans* worm nervous system. Using calcium imaging, it is now possible to map the worm activity onto the structure of the nervous system.[1] Recently, several papers have described oscillatory activity in particular types of motor neurons (namely, classes A and B) [2].[3] We investigate the activity of a different class, class D, which is also hypothesized to display oscillatory activity but has not yet been validated experimentally. In contrast to the previous studies, we aim to image and track all neurons of a single class instead of performing the same procedure separately for each neuron.

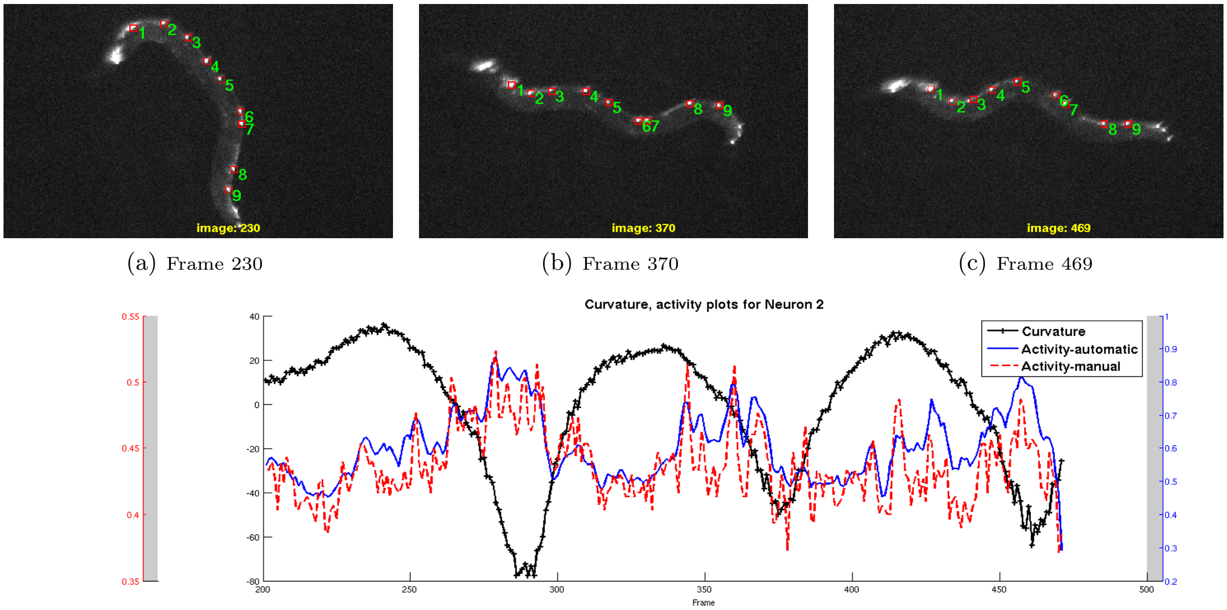
We recorded images of all neurons of class D over a period of time in order to capture multiple activities of the worm such as, moving forward or backward, bending, changing directions etc. Some of these images are displayed in Figures 1(a), (b), (c) at different point of time with neuron locations marked as squares. Ideally, an active neuron region attains a brighter intensity than the inactive ones. Our task is to devise an algorithm that tracks all neurons simultaneously over the sequence. The history of neuron positions produced by the tracking method enables us to compute the curvature and activity for each neuron.

In Figure 1(d), we plot the curvature (black) and activity values (blue, large value implies activity) for one particular neuron, neuron 2 as shown in green in Figures 1(a), (b), (c). This neuron appears to be active just before a dorsal bend in the worm (i.e., just before its curvature starts to increase). The exact correlation between activity and curvature (or other physical phenomenon) is out of the scope of this paper − we describe the tracking mechanism that facilitates such biological analysis in an efficient manner. Manual computation of brightness values for a single neuron over 500 frames requires roughly 1 hour compared to 17 secs (approx) taken by an un-optimized Matlab code without compromising any significant loss in accuracy. The red dashed line in Figure 1(d) corresponds to activity determined by manual tracking and it follows the blue activity curve, generated by automatic tracking, very closely.

Popular single object tracking algorithms in computer vision literature such as feature point tracking,[4] Kalman filter[5] or Particle Filter[6] based tracking, Mean-Shift tracking,[7] discriminative tracking[8] heavily rely on the appearance information to compute a probability (or confidence) of the target to be present around current location in the next frame. In our dataset, the neuron mostly appears to be a bright blob whose shape varies over time and does not offer much information to distinguish it from the background. The variable inter-neuron distances also cause two tracked positions to merge with each other.

Several algorithms have also addressed the task of multiple object tracking where freely moving targets interact with each other in close proximity. These approaches apply nearest neighbor matching,[9] Bayesian

Send correspondence to Toufiq Parag. E-mail: paragt@janelia.hhmi.org



(a) Frame 230

(b) Frame 370

(c) Frame 469

(d) Curvature(black)-activity(blue) plot for Neuron 2 (shown below)

Figure 1. (a), (b), (c) Neurons in different frames (tracked by proposed method). (d) Sample activity curvature plot for Neuron 2, blue: automatic and red: manual tracking.

Network,[10] Linear optimization[11] to correctly match the tracked location to object (primarily human) identities. The common setup of these multi-target tracking methods assumes the targets to wander freely and imposes no restriction on the inter-object distances. On the other hand, in our dataset, the object locations are constrained by the shape of the worm. Utilization of positional constraints renders the sophisticated treatment of object-tracked position association redundant in our scenario.

In the proposed tracking method, we generate a probability map over the search region by filtering the next frame with an isotropic 2D Gaussian (circular blob) template. In each search region (i.e., for each target), several local maxima are identified by a gradient ascent approach same as Mean-shift.[12] These candidate locations are then mapped to the neuron identities by an assignment technique which maximally retains the inter-neuron distances of the past frame. The assignment costs, defined over triplets of matches, constitute a tensor that needs to be factorized in order to solve the matching. We relax the factorization by approximating the cost tensor by a rank-1 tensor whose constituent vector is discretized afterwards to generate integral solutions.

The proposed algorithm can robustly track more neurons than a single object tracking algorithm and a multi-object tracking method based on nearest neighbor assignment in several worm sequences. Our method can in general be applied for tracking sets of objects numbered serially or with a similar shape structure (e.g., tracking a flock of birds[13]) in natural or biological images.

2. CHALLENGES IN MULTIPLE NEURON TRACKING

The neuron appearances in the worm images are not informative in general. They appear to be filled blobs of various shapes that are relatively brighter than the surrounding region. Unlike objects in natural images, the pattern in intensity distribution or some feature arrangement (e.g., histogram of gradients) is absent in these neuron bodies. In Figure 2, we show two neuron bodies (the luminous blobs) along with the corresponding search window. The probability map generated by cross-correlation of neuron patch with the search window and by the Bhattachariya coefficient between their intensity distributions[7] are also shown in Figure 2. None of these probability maps reflects the change in target location and therefore cannot track it successfully.

Isolating brighter neuron area within the template, or search window, is also not possible due to large variation in neuron intensities. The difficulty is further compounded by out of focus blurring, translation of the slide on

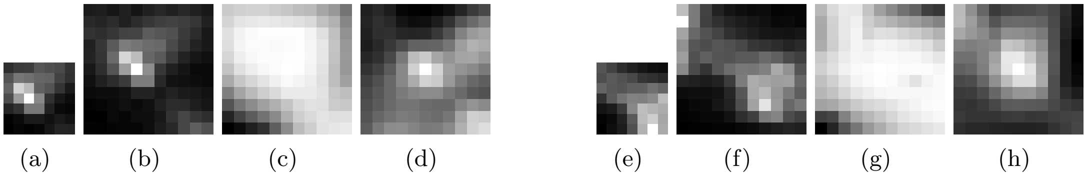


Figure 2. In each four images (a)–(d) and (e)–(f), from left to right: template neuron location, search window, probability maps computed with Bhattachariay coefficient and normalized cross-correlation.

which the animal is placed, sudden reversal in movement direction, proximity of two adjacent neurons (observe, e.g, the neurons 6 and 7 in Figures 1(a), (b), (c)) etc. We illustrate tracking failures due to some of these causes in experiments section.

3. PROPOSED METHOD

Let us suppose $\mathbf{x}_i^t$ is the location of i-th neuron at time t for $i = 1, 2, \ldots, N$. The initial locations $\mathbf{x}_i^0$ are marked manually in a specific order that needs to be preserved throughout the tracking procedure. We first determine several candidate locations $\mathbf{y}_{i,j_i}^{t+1}$ $, j_i = 1, \ldots, M_i^t$ given projected neuron positions $\hat{\mathbf{x}}_i^{t+1}$ and then identify the best match between $\mathbf{y}_{i,j_i}^{t+1}$ and $\mathbf{x}_i^t$ utilizing the shape structure over the set of neurons.

3.1 Candidate Locations

In the first step, the neuron locations in the next frame are approximated with a linear model. Given the tracked positions of neuron i in times t and $t-1$, the projected location is computed by $\hat{\mathbf{x}}_i^{t+1} = \mathbf{x}_i^t + 0.5 \times (\mathbf{x}_i^t - \mathbf{x}_i^{t-1})$.

Next, the image is convolved with an isotropic Gaussian with a standard deviation roughly equal to average neuron width to produce the probability or confidence map . We search for candidate neuron locations $\mathbf{y}_{i,j_i}^{t+1}$ in a square shaped search window $\mathbf{w}_i^{t+1}$, with confidences $C(\mathbf{w}_i^{t+1})$, around $\hat{\mathbf{x}}_i^{t+1}$. These locations are essentially the local maxima within the confidence map and are computed by the Mean-shift mode seeking algorithm[12] assuming continuity over the confidence values. Starting from a specific point $\mathbf{z}^0$, the Mean-shift algorithm iteratively progress towards the direction that maximizes the confidence value until convergence. The iterative update formula can be derived by taking the derivative of a KDE estimate of probability values w.r.t. the current location similar to.[12]

$$\mathbf{z}^{\tau+1} \leftarrow \frac{\sum_{\substack{\mathbf{z}_i \in \mathbf{w}_i^{t+1} \\ v_i \in C(\mathbf{w}_i^{t+1})}} v_i \; \mathbf{z}_i \exp\left(\frac{-||\mathbf{z}_i - \mathbf{z}^\tau||^2}{h^2}\right)}{\sum_{v_i \in C(\mathbf{w}_i^{t+1})} v_i \; \exp\left(\frac{-||\mathbf{z}_i - \mathbf{z}^\tau||^2}{h^2}\right)} \tag{1}$$

In order to determine the local intensity maxima $\mathbf{z}_{i,l}^*, l = 1, \ldots, L_i^{t+1}$, multiple Mean-shift search is performed in each window $\mathbf{w}_i^{t+1}$ starting from all locations having confidences larger than a threshold . For all our experiments, this threshold was set to mean$(C(\mathbf{w}_i^{t+1}))$ + std-dev$(C(\mathbf{w}_i^{t+1}))$. Ideally Mean-shift procedure initiated from several nearby locations would converge to the same local maxima. Let us denote the number of initial positions that converge to any local maxima $\mathbf{z}_{i,l}^*$ as its support $\psi_{i,l}$. For any i, all local maxima with a support $\psi_{i,l} > \delta$ are clustered together and their centers are selected to be the candidate locations $\mathbf{y}_{i,j_i}^{t+1}$.

3.2 Neuron Assignment

Let us use a shorthand notation $\{i, j_i\}$ for any potential matching pair $\{\mathbf{x}_i^{t+1}, \mathbf{y}_{i,j_i}^{t+1}\}$ and let α_{i,j_i} indicate whether the match is correct ($\alpha_{i,j_i} = 1$) or incorrect ($\alpha_{i,j_i} = 0$). We wish to utilize the quality of tentative matches of three consecutive neurons in the same order they are numbered, e.g., the order shown in Figures 1(a), (b), (c). The matching score is quantified as how much the inter-neuron distances, from the last known locations in time t, are preserved by the candidate locations in time $t+1$. To this end, a matching cost is defined on the deviance between the pairwise distances of two pairs of neurons in the last frame and those between the candidate matches in the current frame.

$$\phi_b(\{i-1, j_{i-1}\}, \{i, j_i\}, \{i+1, j_{i+1}\}) = \| \; [d_b(x_i^t, x_{i-1}^t) \quad d_b(x_{i+1}^t, x_i^t)]^T - [d_b(y_{j_i}^{t+1}, y_{j_{i-1}}^{t+1}) \quad d_b(y_{j_{i+1}}^{t+1}, y_{j_i}^{t+1})]^T \; \|^2 \tag{2}$$

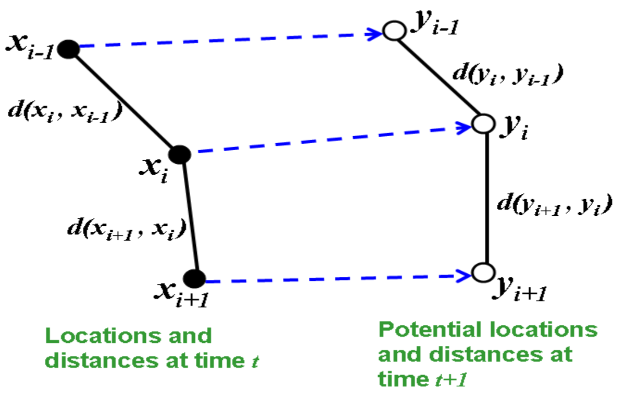


Figure 3. Computation of $\phi_r(\{i-1, j_{i-1}\}, \{i, j_i\}, \{i+1, j_{i+1}\})$ for neuron matching by preserving inter-neuronal distances. Left: target location in frame t, right: candidate locations at time $t+1$. Blue dashed lines indicate tentative assignment.

In Equation 2, $b \in \{r, \theta\}$ corresponds to two modes of distances, spatial (Euclidean) and angular. The first vector $[d_b(x_i^t, x_{i-1}^t) \;\; d_b(x_{i+1}^t, x_i^t)]^T$ in the cost computes the differences in the consecutive locations in time t and the the second vector holds those between the corresponding matches in time $t+1$ as shown in Figure 3. For simplicity in notation, let us use 1D indices k for the matches $\{i, j_i\}$. The spatial and angular costs are combined to form the following match quality score expressed in terms of 1D indices.

$$q(k_1, k_2, k_3) = \exp\Big(-\frac{\phi_r(k_1, k_2, k_3)}{\sigma_r^2} - \frac{\phi_\theta(k_1, k_2, k_3)}{\sigma_\theta^2}\Big) \tag{3}$$

With these ternary matching scores, the optimal values of $\boldsymbol{\alpha}$ would maximize the inner product between the matching score tensor and $\boldsymbol{\alpha} \circ \boldsymbol{\alpha} \circ \boldsymbol{\alpha}$ where $\circ$ imply the outer product.

$$\arg\max_{\alpha} \sum_{\alpha_{k_1}, \alpha_{k_2}, \alpha_{k_3}} \alpha_{k_1}\alpha_{k_2}\alpha_{k_3} \; q(k_1, k_2, k_3). \tag{4}$$

The integrality constraints on α_k are relaxed to assume real values in our attempt to solve the tensor maximization problem. We use a rank-1 approximation of the matching tensor, computed by higher order Power method,[14] as a solution of this optimization. In this method, each vector is updated by the inner product of two sub-tensors resulted from keeping the vector's dimension fixed. The cost tensor in our framework is substantially different from that in,[15] in terms of both the definition (Equations 2 and 3) and structure. The cost tensor defined in Equations 2 and 3 is asymmetric and therefore the decomposition algorithms in,[15] which assumes symmetry, can not be readily applied here.

In all our experiments, only 5 iterations of the Power method produced a real-valued indicator vector informative enough to distinguish the accurate assignments from the inaccurate ones. To compute a discrete decision from these real numbers, for each neuron i, we find out the largest entry in vector $\boldsymbol{\alpha}$ among the ones that correspond to this neuron and and assign it the matching location. The resulting matching location becomes $\mathbf{x}_i^{t+1}$.

4. EXPERIMENT AND RESULTS

We report the performances of the proposed tracking method on 4 image sequences with more than 400 images each. Each sequence contains different number of neurons that are manually selected in the first frame. We used the following parameters values for all sequences: support and standard deviation of Gaussian filter 5 and 1.667, support threshold $\delta = 5$, bandwidth for mean-shift profile function $= 0.75$, and $\sigma_r = 0.01$, $\sigma_\theta = 0.1$ in the definition of ternary cost in Equation 2.

We compared the performance of our method to those of two tracking algorithms: 1) Single object tracking (MS): runs Mean-shift procedure on the probability image generated by the Gaussian filter; and 2) Multi-object tracking algorithm (HM) similar to:[9] applies the Hungarian Matching technique to select the candidate $\mathbf{y}_{i,NN}^{t+1}$ closest to $\mathbf{x}_i^t$ as new target location $\mathbf{x}_i^{t+1}$ instead of the shape-preserving assignment described in Scetion 3.2.

Table 1. Performance comparison of the tracking algorithms in terms of percentage of neurons accurately tracked.

Sequence	Frames	Neurons	MS	HM	Proposed
Seq 1	435	14	64.29	57.14	**85.71**
Seq 2	400	14	85.71	71.43	**92.86**
Seq 3	450	14	50.00	57.14	**71.43**
Seq 4	550	13	61.54	*76.92*	*76.92*

The set of parameters that overlaps with the proposed approach are set to the same values as above, other parameters are tuned to generate the best result. We also tested the Kanade-Lucas-Tomasi (KLT) feature tracker (http://www.ces.clemson.edu/~stb/klt/) on our data. The performance of KLT tracker is too poor to be compared with the aforementioned algorithms and we exclude its results from our analysis. Table 1 shows the percentage of neurons that were successfully tracked throughout the sequence. In case of a merge, we count only the neuron that remained in its correct location. We provide the videos (with enhanced brightness/contrast) for these tracking results, including the output of KLT, in supplementary material.

Let us examine how the proposed method avoids merges of two nearby neurons that the MS method is susceptible to. Figure 4(a) shows a search region for a particular neuron (index 4) of Sequence 2 (frame 146). The search region contains two local maxima: the one near the center of the search window is the actual location of neuron 4. The peak at the upper left corner belongs to another neuron (index 3) and appears here due to close proximity. Our candidate selection technique (Section 3.1) was able to identify both the local maxima as shown in Figure 4(b). In the following stage of assignment, neuron 4 was accurately assigned to the maximum at the center. Notice that, the blob near the center is dimmer, i.e., contains a maximum with lower intensity value than that situated at the upper left corner. That is why mean-shift (MS) procedure, designed for searching a single peak, converges at the upper-left corner thereby merging neuron 4 to the location of neuron 3 (Figure 4(c)).

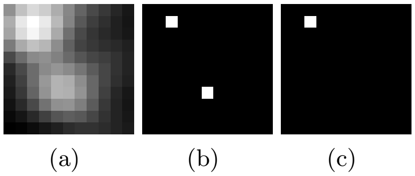


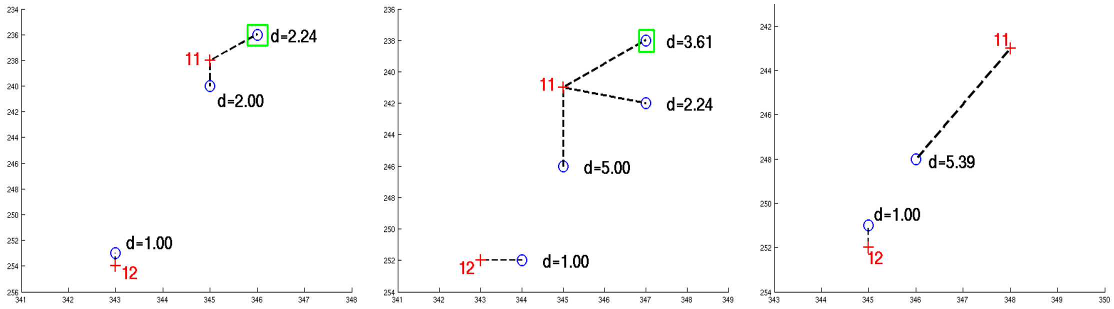
Figure 4. (a): search window, blob near the center is the correct location . (b) shows two maxima located by our method (white squares) and (c) false maximum computed by single-mode mean-shift.

Figure 5 illustrates a failure case for the nearest neighbor based approach HM in frames 212–214 of Sequence 2. The red '+' indicate locations $\mathbf{x}_{11}^t, \mathbf{x}_{12}^t$ of two neurons respectively in past frame, the blue circles point to the candidate locations $\mathbf{y}_{11,j_{11}}^{t+1}, \mathbf{y}_{12,j_{12}}^{t+1}$ and the green squares indicate the actual locations of neurons 11 and 12 in current frame. The distances from target locations in previous frame (red +) to the corresponding candidates (blue o) are also mentioned in these plots. It is obvious that the nearest candidate search would not yield the actual neuron locations which is why HM gradually shifts the tracked position of neuron 11 towards neuron 12 (observe that in Figure 5(c) there is only one potential position for neuron 11 which is not correct) and merges the two thereafter. HM will be even more vulnerable when the worm starts moving to the opposite direction possibly bringing the candidates of preceding or following neurons closer to any target than those of its own. The position that maximally conforms to the worm shape, rather than the brightest or nearest ones, is more likely to represent the actual location of a neuron. Our approach encodes shape information in terms of inter-neuron distances and has been capable of tracking more neurons correctly so far.

In Figures 6(a), (b), we show 1D tracked locations of a subset of neurons (index 2 – 6) in Sequence 3 by MS, HM and the proposed method respectively. Our tracking algorithms was able to track 4 out of 5 neurons over the whole sequence whereas both MS and HM merged 2, 3, 4 and 5, 6 in very early stage of tracking (around frame 75) due to a sudden reverse movement and slide translation. These plots would assist the user to quickly identify tracking failures, if any, in long sequences.

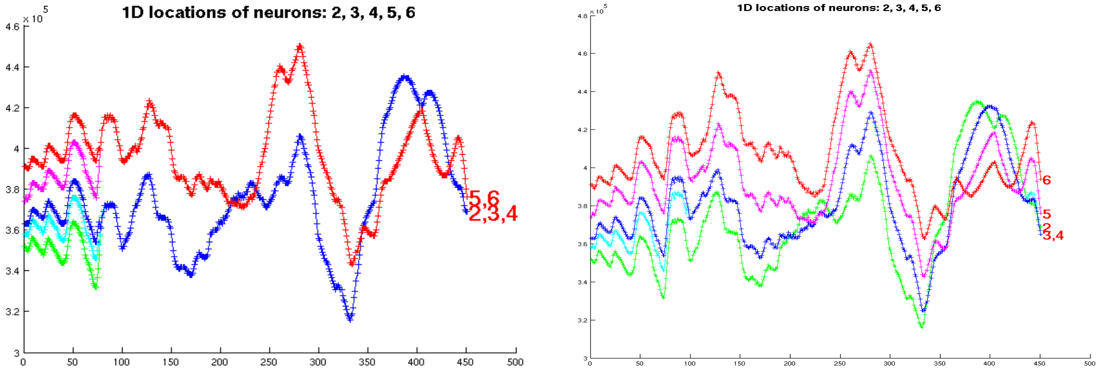
5. CONCLUSION

This paper proposes a robust multi-neuron tracking algorithm for long sequence of images. Due to appearance ambiguity, our method computes several potential locations for each nueron. Then, a matching algorithm associates each neuron to its best match maintaining the worm shape structure. Experimental results presented



(a) Frame 212 (b) Frame 213 (c) Frame 214

Figure 5. Tracking failure of nearest neighbor based method HM.



(a) MS and HM (b) Propsoed

Figure 6. 1D tracked locations for neurons 2–6 in Seq 3.

in the paper demonstrate the superiority of the proposed method over both a single object and a multi-object tracking algorithm utilizing nearest neighbor match. In addition, our method has already tracked, with higher success rate than the aforementioned methods, neurons in more than 6000 frames that has not been reported in this paper. We believe this work can be extended to address other tracking challenges in biological images for objects with a specific shape structures.

REFERENCES

[1] Kerr, R., Lev-Ram, V., Baird, G., Vincent, P., Tsien, R. Y., and Schafer, W. R., "Optical imaging of calcium transients in neurons and pharyngeal muscle of C. Elegans," *Neuron* **26**(3), 583 – 594 (2000).

[2] Faumont, S., Rondeau, G., Thiele, T. R., Lawton, K. J., McCormick, K. E., Sottile, M., Griesbeck, O., Heckscher, E. S., Roberts, W. M., Doe, C. Q., and Lockery, S. R., "An image-free opto-mechanical system for creating virtual environments and imaging neuronal activity in freely moving *Caenorhabditis elegans*," *PLoS ONE* **6**(9), e24666 (2011).

[3] Kawano, T., Po, M., Gao, S., Leung, G., Ryu, W., and Zhen, M., "An imbalancing act: Gap junctions reduce the backward motor circuit activity to bias c. Elegans for forward locomotion," *Neuron* **72**(4), 572 –586 (2011).

[4] Shi, J. and Tomasi, C., "Good features to track," in [*CVPR*], (1994).

[5] Dellaert, F. and Thorpe, C., "Robust car tracking using kalman filtering and bayesian templates," in [*Conference on Intelligent Transportation Systems*], (1997).

[6] Isard, M. and Blake, A., "Condensation – conditional density propagation for visual tracking," *IJCV* **29**, 5–28 (1998).

[7] Comaniciu, D., Ramesh, V., and Meer, P., "Kernel-based object tracking," *PAMI* **25**, 564–575 (May 2003).

[8] Avidan, S., "Ensemble tracking," *PAMI* **29**(2), 261 –271 (2007).

[9] Perera, A., Srinivas, C., Hoogs, A., Brooksby, G., and Hu, W., "Multi-object tracking through simultaneous long occlusions and split-merge conditions," in [*CVPR*], (2006).

[10] Nillius, P., Sullivan, J., and Carlsson, S., "Multi-target tracking - linking identities using bayesian network inference," in [*CVPR*], (2006).

[11] Ben Shitrit, H., Berclaz, J., Fleuret, F., and Fua, P., "Tracking Multiple People under Global Appearance Constraints," in [*ICCV*], (2011).

[12] Comaniciu, D. and Meer, P., "Mean shift: a robust approach toward feature space analysis," *PAMI* **24**, 603 –619 (may 2002).

[13] Chen, W., Liu, H., Hu, S., and Ning, H., "Group tracking of flock targets in low-altitude airspace," in [*Parallel and Distributed Processing with Applications Workshops (ISPAW)*], (2011).

[14] Lathauwer, L. D., Comon, P., Moor, B. D., , and Vandewalle, J., "Higher-order power method – application in independent component analysis," in [*Int. Symp. Nonlinear Theory Application*], (1995).

[15] Duchenne, O., Bach, F., Kweon, I., and Ponce, J., "A tensor-based algorithm for high-order graph matching," in [*CVPR*], (2009).