A Global Spatial Similarity Optimization Scheme to Track Large Numbers of Dendritic Spines in Time-Lapse Confocal Microscopy

Qing Li, Zhigang Deng*, Yong Zhang, Xiaobo Zhou, U. Valentin Nägerl, and Stephen T. C. Wong*

***Abstract*—Dendritic spines form postsynaptic contact sites in the central nervous system. The rapid and spontaneous morphology changes of spines have been widely observed by neurobiologists. Determining the relationship between dendritic spine morphology change and its functional properties such as memory learning is a fundamental yet challenging problem in neurobiology research. In this paper, we propose a novel algorithm to track the morphology change of multiple spines simultaneously in time-lapse neuronal images based on nonrigid registration and integer programming. We also propose a robust scheme to link disappearing-and-reappearing spines. Performance comparisons with other state-of-the-art cell and spine tracking algorithms, and the ground truth show that our approach is more accurate and robust, and it is capable of tracking a large number of neuronal spines in time-lapse confocal microscopy images.**

***Index Terms*—Dendritic spine, free form deformation, global similarity, integer programming, time-lapse images.**

I. Introduction

In neurobiology, dendrites are tree-like structures situated at the beginning of neuronal cells whose main purpose is to help increase the surface area of the cell body, covered with synapses. Spines are small protrusions on the surface of a dendrite which receive excitatory synaptic input from other neurons and transmit electrical stimulation to the soma. The rapid and spontaneous morphology changes of dendritic spines have been widely observed [15], [25], [35]. Various spine shapes are classified into several categories, e.g., mushroom, thin, stubby, etc. Existing neurobiology literature [17], [27], [30] shows that the morphological plasticity of dendritic spines is highly correlated with their underlying cognitive functionality (e.g., sensory experience, learning, and memory). Therefore, how to efficiently and accurately extract and track the morphological features of spines in time-lapse microscopy neuron images is crucial to study roles of spines related to those cognitive functionalities.

Proper understanding of the mechanism and pathway of various neurological conditions requires accurate spine contours. Currently, spine contours and spine association across time are obtained using human-assisted software. Considering that a single branch of dendrite may contain hundreds of spines, manually segmenting and establishing the spine correspondences in time-lapse images is tedious, and is subject to human bias. Automatically extracting and tracking the morphology features of spines is highly desired as it would provide consistent results and reduce the workload of neurobiologists.

Automated segmentation and tracking of dendrite spines is challenging, because: 1) Spines have small volumes ranging typically from 0.01 μm^3 to 0.8 μm^3. It challenges both the resolution of the confocal microscope and the accuracy of the segmentation algorithm. 2) Spines have highly variable visual shapes and can appear anywhere on the surface of a dendrite. This makes it nearly impossible to create templates or use any prior shape information to detect, segment, and track spines. 3) Existing literature [15], [25], [35] shows that changes in the shape of spines can be dramatic (e.g., in certain cases, spines could completely disappear for certain periods). Therefore, it is difficult to use spine shape as a consistent feature for tracking.

In this paper, we propose a novel, automated algorithm to track dendritic spines individually in time-lapse images. Its core idea is to maximally utilize the spatial and structural information between two dendritic structures as follows. Given two dendritic structures at two consecutive time points, global registration is first applied to correct possible translational and rotational shift of the specimen during image acquisition. Then, nonlinear local shape deformation and the optimization of global similarity metrics are introduced to establish temporal correspondences between two dendritic structures. Finally, through the analysis of spine centroid trajectories, invalid associations are discarded. In addition, we also propose a novel scheme to address the spine disappearing-and-reappearing

Manuscript received July 17, 2010; revised September 13, 2010 and October 24, 2010; accepted October 24, 2010. Date of publication November 01, 2010; date of current version March 02, 2011. This work was supported in part by a Bioinformatics Program grant from The Methodist Hospital Research Institute of Weill Cornell Medical College, John S Dunn Foundation Distinguished Endowed Chair, NIH R01 LM009161 to STCW, Texas NHARP 003652-0058-2007, and NSF IIS-0914965. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies. *Asterisk indicates corresponding author.*

Q. Li is with Computer Science Department, University of Houston, Houston, TX 77004 USA.

*Z. Deng is affiliated with Computer Science Department, University of Houston, Houston, TX 77004 USA (e-mail: zdeng@cs.uh.edu).

Y. Zhang was with Methodist Hospital Research Institute, Houston, TX 77030 USA. He is now with Healthcare Informatics, IBM Almaden Research Center, San Jose, CA 95120 USA.

X. Zhou is with the Center for Biotechnology and Informatics, the Methodist Hospital Research Institute, Houston, TX 77030 USA.

*S. T. C. Wong is with the Center for Biotechnology and Informatics, the Methodist Hospital Research Institute, Houston, TX 77030 USA (e-mail: stwong@tmhs.org).

U. Valentin Nägerl is affiliated with Victor Segalen Bordeaux 2 University, 33076 Bordeaux, France.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMI.2010.2090354

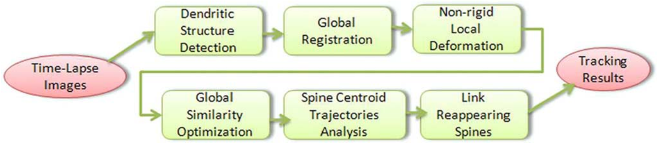
Time-Lapse Images
Dendritic Structure Detection
Global Registration
Non-rigid Local Deformation
Global Similarity Optimization
Spine Centroid Trajectories Analysis
Link Reappearing Spines
Tracking Results

Fig. 1. Pipeline of our dendritic spine tracking algorithm.

issue by analyzing local spatial relationships of spine candidate association pairs through graph matching. It is noteworthy that our algorithm can be straightforwardly extended to 3D, if the spatial resolution along the Z axis of the acquired microscopy data can be improved. The pipeline of our approach is illustrated in Fig. 1.

The major contributions of this work include: 1) a robust scheme to track a large number of temporally-varying spines simultaneously through dendrite-specific, nonlinear deformation and global similarity optimization, and 2) a novel scheme to address the challenging spine disappearing-and-reappearing issue for dendritic spine tracking, by analyzing local spatial relationships of spine candidate association pairs.

The remainder of this paper is organized as follows: related work is reviewed in Section II. Image acquisition and preprocessing are described in Section III. Dendritic structure detection is presented in Section IV, followed with dendritic structure alignment in Section V and spine association determination in Section VI. Experimental results are detailed in Section VIII, and Section IX concludes our work with discussion and future directions.

II. Related Work

Robust object tracking is a long-standing research topic in computer vision [38]. Its goal is to estimate the trajectories of moving objects and solve their associations over time. Based on their underlying principles, existing algorithms can be categorized into two groups: *bottom-up* approaches and *top-down* approaches. In the bottom-up approaches [1], [7], [21], [22], [28], [34], [37], objects are first detected and represented in a feature space, and then temporal associations are solved by maximizing a similarity function. In the top-down approaches [2], [6], [9], boundaries or positions of objects are initialized at the first frame. Then, boundaries and positions of objects are evolved frame by frame through a prior statistical model. Our dendritic spine tracking algorithm falls into the bottom-up category. Typically, the suitability of a particular tracking algorithm depends on many factors such as object appearance, number of objects, and object and camera motions.

A few existing efforts have been attempted to accurately track dendritic spines. For example, Koh *et al.* [18] proposed a simple and intuitive approach to track dendritic spines. In their approach, after global registration, two spines in different frames are considered to be the same, if their overlapping area is larger than 25% of the volume of either of them. However, if in certain areas the spine density is high, their approach may yield incorrect one-to-many correspondences. In the work of [13] and [26], a graph based matching model was proposed to track dendritic spines. Each node in the graph represents a spine branch and is connected with its neighbors. A six-dimensional Gaussian distribution is employed as the probability distribution function to determine whether two nodes in different graphs represent the same spine. The mapping between two graphs is obtained by maximizing a posterior probability. However, whether the spine spatial and temporal distribution follows the Gaussian distribution is not justified in their paper. In addition, their approach requires users to manually tune its critical parameters in its Gaussian model, which affects its robustness and automation.

III. Image Acquisition and Pre-Processing

Time-lapse two-photon laser scanning fluorescence microscopy was used to monitor the morphological plasticity of dendritic spines of CA1 pyramidal neurons in organotypic hippocampal brain slice preparation. The term CA1 (cornu ammonis) refers to an anatomically defined region within the hippocampus, where the CA1 pyramidal neurons are located, which are the main excitatory neurons in this area. To this end, individual neurons were briefly loaded via a patch pipette containing a green fluorescent dye [4 mm calcein green (Invitrogen)]. The field of view for the imaging was chosen within the stratum oriens of the CA1 pyramidal neurons, that is, a histologically defined area on the basal side of this neuronal cell population. 3D image stacks (spanning 140 μm in x, 140 μm in y, and 25–40 μm in z, 1024 × 1024 pixels in xy and 0.5 μm step size in z) were acquired every 30 min for up to 6 h. The green fluorescent label did not bleach significantly over the duration of the experiment. Full experimental details can be found in the previous publication [3]. Our neuron images were acquired in 3D (60 image sections in total) to ensure a significant part of the dendrite were contained in the image stack. The 3D image stacks were condensed into 2D images by a maximum intensity projection (MIP), facilitating image handling and data analysis. To ensure MIP is permissible we examined all individual image sections to make sure that different stretches of dendrite did not get mixed up in the projection. With regards to individual spines, the optical sectioning by two-photon microscopy represents a local projection because the size of spines is typically less than the spatial resolution along the Z axis of the microscope. Therefore, in this paper, all spines are detected and tracked in MIP images. The dataset can be provided upon request for research purpose.

IV. Dendritic Structure Detection

Several algorithms have been proposed for dendritic spine detection [8], [23], [39], [40]. In this work, dendrite backbones and spine boundaries, along with branch points in different time points are detected using a curvilinear structure detection based algorithm [39]. In this algorithm, line segments with similar local directions are first obtained through the analysis of the largest eigen-value and its corresponding eigen-vector of the Hessian matrix. If two estimated line segments are connected in an eight-connected neighborhood or the distance between them is smaller than a prespecified linking threshold, the two line segments will be linked. Spine backbones and dendrite backbone are distinguished by their lengths. Branch points are detected as the pixels that have three or more neighbors. Fig. 2 shows the

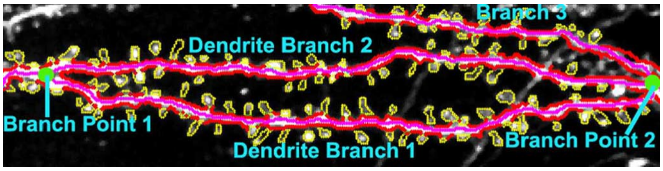

Fig. 2. The dendritic structure detection results of an example image. Spine boundaries are labeled in yellow lines, the pink lines illustrate the dendritic backbones, and the dendritic boundaries are labeled in red line. There are three dendritic branches in this image. The green dots illustrate the branch points.

detected dendritic structures in an example image. The boundaries of spines are highlighted in blue and the backbones of the dendrite are drawn in pink. We assume that color printing is used for retaining the information in color figures in this paper.

V. Dendritic Structure Alignment

After dendritic structures are detected, the spine boundaries and dendrite backbones are represented as 2D shapes. In this section, determining spine associations between two consecutive frames is explained in details. First, in Section V-A, global registration is adopted to correct the image data shift during the image acquisition stage. How to design a nonrigid deformation to optimally align two dendritic structures is described in Section V-B. In Section VI, spine associations are determined through global spatial similarity maximization after the dendritic structure alignment.

A. Global Registration

Although the best effort was attempted to reposition the specimen at the same place each time during the image acquisition stage, misalignment is inevitable due to human error and system bias. Therefore, a global registration is needed to correct the image data shift between different time points. In this work, the iterative closest reciprocal point (ICRP) algorithm [29] is employed for this purpose. ICRP is a variant of iterative closest point (ICP) algorithm [5] that assumes each point on the source shape has a valid correspondence on the target shape. This assumption is not applicable when two shapes are partially overlapping. To overcome this weakness, ICRP [29] uses ϵ-reciprocal correspondence to reject wrong pairs. During ICRP, a rigid transformation matrix is estimated by minimizing the squared error of closest points between two geometric shapes. The ICRP algorithm is composed of two steps: generating temporary correspondences and estimating the rigid body transformation. The solution is found when the change in mean square error falls below a threshold, which specifies the required precision of the registration. Fig. 3 shows example results of the global registration.

B. Nonrigid Local Deformation

Usually, the morphological changes of spines are rapid, nonrigid, and independent of one another. Therefore, global registration alone is insufficient to align two dendritic structures. This stands true especially at locations where spine density is high. Inspired by the free form local registration algorithm [16], [31], in this work, the dendritic structure at time point t_{k+1} (i.e., the source shape) is locally deformed to match the dendritic structure at time point t_k (i.e., the target shape) by minimizing a specially-designed, dendritic structure-incorporated energy function. Point-to-point correspondences can be established after the local deformation. Finally, spine associations between different time points are solved through the global maximization of a spatial similarity metric.

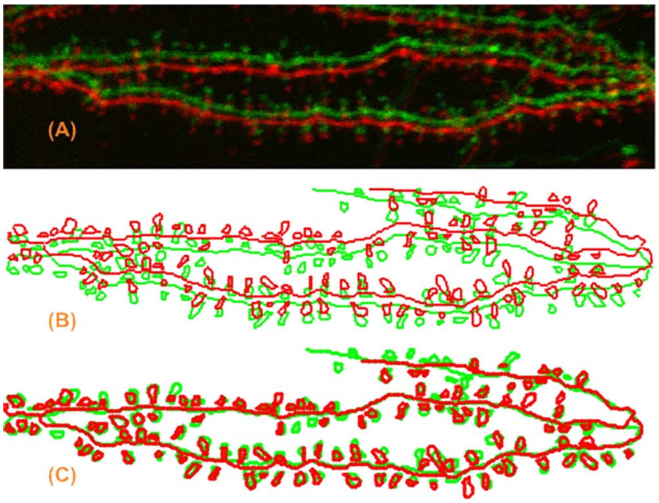

Fig. 3. Example results of the global registration. (a) The original source image and target image are displayed in green and red colors, respectively. (b) The extracted source dendritic structure (red) and the extracted target dendritic structure (green). (c) After global registration, the source structure (red) rotates and translates to match with the target structure (green).

In our algorithm, the source and target shapes are first transformed to an implicit shape representation. The transformation is described as follows. Let S be a shape, i.e. a curve in 2D, and $\Phi : \Omega \rightarrow R^+$ be a Lipschitz function that represents a distance transform of shape S, Ω is the image domain, the region enclosed by S is R_S, and the background region is $\Omega \setminus R_S$. The implicit shape representation is defined as follows:

$$\Phi_S(x,y) = \begin{cases} 0, (x,y) \in S \\ +\mathrm{d}\left((x,y),S\right) > 0, \quad (x,y) \in R_S \\ -\mathrm{d}\left((x,y),S\right) < 0, \quad (x,y) \in |\Omega \setminus R_S| \end{cases} . \quad (1)$$

Here, $\mathrm{d}((x,y),S)$ calculates the minimum Euclidean distance between an image pixel located at (x,y) and shape S. Fig. 4(a) shows the visualization of the source shape in the implicit shape representation. The advantage of this implicit representation is that gradient descent methods can be conveniently applied to it, and it can facilitate the imposition of smoothness constraints on the shape boundary.

Local registration is equivalent to recovering a pixel-wise local deformation field that creates correspondences between the target shape and the source shape. Such a local deformation field can be efficiently represented using space warping models [12], [16], [19], [20], [31]. We choose the free form deformation (FFD) model [16], [31] in this work, because it is capable of implicitly enforcing smoothness constraints, preserving shape topology, and guaranteeing one-to-one mapping. The essence of FFD is to deform the source shape by manipulating a regular control lattice overlaid on its space to minimize the difference between the source shape and the target shape. The deformation

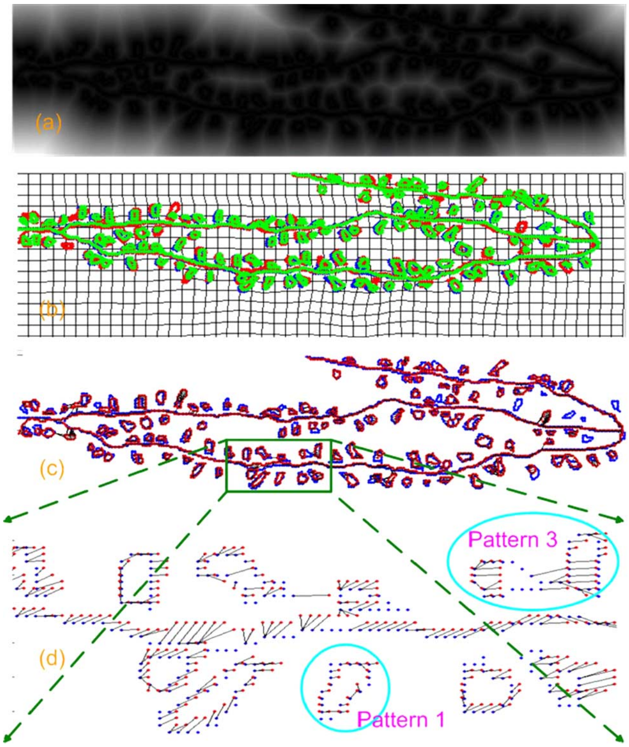

Fig. 4. Local nonrigid deformation: the source shape (blue), the target shape (red), and the deformed shape (green). (a) Visualization of the source shape in the implicit shape representation. (b) Visualization of the deformation filed. (c) The one-to-one pixel correspondences between the source and the deformed source shape. (d) A close view of the rectangular region in (c).

of every pixel on the source shape can be computed through interpolating *B*-spline or Bezier basis functions, described below.

Let $\mathbf{\Theta}_{m,n} = \{(\mathbf{\Theta}^x_{m,n}, \mathbf{\Theta}^y_{m,n}); m = 1,\ldots,M; n = 1,\ldots,N\}$ be a lattice of points overlaid to a region in the embedding space enclosing the source shape region $\phi = \{(x,y)|1 \le x \le X, 1 \le y \le Y\}$. M and N denotes the number of points used horizontally and vertically. The deformed position of any pixel $\xi = (x,y)$ in the embedding space is calculated as a cubic *B*-spline interpolation on the lattice, as follows:

$$\mathrm{D}(\mathbf{\Theta},\xi) = \sum_{k=0}^{3}\sum_{l=0}^{3} \mathrm{B}_k(u)\mathrm{B}_l(v)\mathbf{\Theta}_{i+k,j+l}. \tag{2}$$

Here $\mathbf{\Theta}_{i+k,j+l}$, $(k,l) \in [0,3] \times [0,3]$ (conceptually like "weights") are coordinates of the 16 points on the lattice in the neighborhood of the pixel $\xi = (x,y)$ in Ω, where $i = \lfloor (x/X)\cdot(M-1)\rfloor + 1$, $j = \lfloor (y/Y)\cdot(N-1)\rfloor + 1$. $\mathrm{B}_k(u)$ and $\mathrm{B}_l(v)$ represent the kth and lth cubic *B*-spline basis functions, respectively, where $u = (x/X)\cdot M - \lfloor (x/X)\cdot M\rfloor$, $v = (y/Y)\cdot N - \lfloor (y/Y)\cdot N\rfloor$.

The local registration is equivalent to finding the lattice $\mathbf{\Theta}$ that minimizes the sum of squared differences (SSD) between the implicit (distance transform of) representation of the target and source shapes. Therefore, a corresponding energy function can be defined as follows:

$$\mathrm{E}_{\mathrm{SSD}}(\mathbf{\Theta}) = \iint_\Omega \left[\Phi_T(\xi) - \Phi_S\left(\mathrm{D}(\mathbf{\Theta},\xi)\right)\right]^2 d\xi. \tag{3}$$

In order to preserve the regularity of the deformed shape, a smoothness term is added [(4)]

$$\mathrm{E}_{\mathrm{Smooth}}(\mathbf{\Theta}) = \iint_\Omega \left(\left\|\frac{\partial \mathrm{D}(\mathbf{\Theta},\xi)}{\partial x}\right\|^2 + \left\|\frac{\partial \mathrm{D}(\mathbf{\Theta},\xi)}{\partial y}\right\|^2\right) d\xi. \tag{4}$$

We also define a repulsive constraint to avoid misalignments between dendritic branches on the source shape and those on the target shape. In the detection step, dendrites are divided into branches and assigned to distinct labels. For example, if there are N branches in an image, they will be labeled as $1,2,3,\ldots,N$. Spines will have the same label as the dendrite branch they belong to. Therefore, each pixel on the dendritic structure (both the dendrite backbone and spines) is assigned a label based on its association with the dendritic branches. During the deformation, we want to keep the number of misaligned pixels on the source shape as small as possible, that is, we want pixels labeled as i on the source shape to be aligned as pixels labeled as i on the target shape.

The repulsive constraint can be formalized as follows:

$$\mathrm{E}_{\mathrm{Rep}} = \sum_{i=1}^{N}\sum_{j=1,j\ne i}^{N} \left|C_i^T \cap C_j^S\right|. \tag{5}$$

Branch labels are assigned to pixels in shapes. Here, $\{C_j^S | j = 1,2,\ldots,N\}$ denotes dendritic branch labels on the source shape and $\{C_i^T | i = 1,2,\ldots,N, i \ne j\}$ denotes dendritic branch labels on the target shape. The "branch labels" are integer numbers from 1 to N (N is the total number of the branches), and they are assigned to pixels in shapes. Note that the source shape and the target shape have the same number (N) of dendritic branches, because the dendrites are stable during the image acquisition. $\mathrm{E}_{\mathrm{Rep}}$ returns how many pixels on the source shape are misaligned with different branch structures.

Finally, by combining all the above three terms [(3)–(5)] together, we obtain a summed energy functional E for the nonrigid registration of two dendritic structures:

$$\mathrm{E}(\mathbf{\Theta}) = \mathrm{E}_{\mathrm{SSD}}(\mathbf{\Theta}) + \alpha\mathrm{E}_{\mathrm{Smooth}}(\mathbf{\Theta}) + \omega\mathrm{E}_{\mathrm{Rep}}. \tag{6}$$

Here α is a constant parameter to control the smoothness of the deformed shape, and ω is a positive parameter to control the repulsive constraint between different branches. Specifically, α and ω are experimentally set to 2.0 and 5.5 in our experiments, respectively. These parameters may vary for different datasets to achieve better results. Generally, if the morphology of spines changes rapidly in an image dataset, the smoothness of spine shape is not our first concern. Therefore, α can be set to a small value. If the spine density in the branch region is high, to prevent spines that belong to different branches being aligned together, ω should be set to a higher value.

Then, a gradient descent method is used to optimize the criterion defined by (6). Its partial derivative is computed as follows:

$$\frac{\partial}{\partial\theta_i}\mathrm{E}(\mathbf{\Theta}) = -2\iint_\Omega \left[\Phi_T(\xi) - \Phi_S\left(\mathrm{D}(\mathbf{\Theta};\xi)\right)\right] \times\left(\nabla\Phi_S\left(\mathrm{D}(\mathbf{\Theta},\xi)\right)\cdot\frac{\partial}{\partial\theta_i}\mathrm{D}(\mathbf{\Theta},\xi)\right) d\mathbf{x}$$

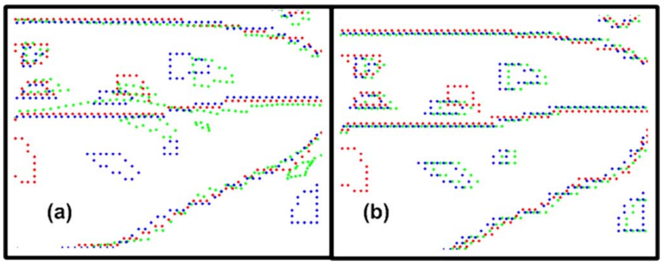

Fig. 5. Nonrigid local deformation comparison with/without using the repulsion constraint. The source shape is in red, the target shape is in blue, the deformed shape is in green. (a) The green dots show the deformed shape without using the repulsion constraint. (b) The green dots show the deformed shape after using the repulsion constraint.

$$+2\alpha \iint_{\Omega} \frac{\partial \mathrm{D}(\boldsymbol{\Theta}, \xi)}{\partial x} \cdot \frac{\partial}{\partial \theta_i} \left(\frac{\partial}{\partial x} \mathrm{D}(\boldsymbol{\Theta}, \xi) \right) + \frac{\partial \mathrm{D}(\boldsymbol{\Theta}, \xi)}{\partial y} \cdot \frac{\partial}{\partial \theta_i} \left(\frac{\partial}{\partial y} \mathrm{D}(\boldsymbol{\Theta}, \xi) \right) d\xi + \omega \frac{\partial}{\partial \theta_i} \mathrm{E}_{\mathrm{rep}}. \tag{7}$$

Here $(\partial/\partial\theta_i)\mathrm{E}_{\mathrm{Rep}}$ is calculated as $\mathrm{E}_{\mathrm{Rep}}(\boldsymbol{\Theta}_k) - \mathrm{E}_{\mathrm{Rep}}(\boldsymbol{\Theta}_{k-1})$. Note k denotes the iteration number and this partial derivative is calculated in the lattice points' difference space, not in time space. After the deformation, we establish pixel correspondences between the source shape and the target shape. For each pixel on the deformed source shape, we calculate its nearest point on the target shape as the correspondence point. Note that the dense one-to-one mapping feature in FFD denotes the pixel correspondence between the source shape and the deformed source shape; the established correspondence between the source shape and the target shape may not be one-to-one correspondence. However, for each pixel on the source shape, only one correspondence can be found on the target shape. Fig. 5 shows a nonrigid local deformation comparison with/without using the repulsion constraint. From Fig. 5, we can see that the repulsion force prevents pixels belong to the different branches merging together.

Based on the local nonrigid deformation outcomes, we found that three different pixel association patterns existed. 1) All the pixels of a spine p_a on the target shape are associated with all the pixels of another spine p_b on the source shape. It suggests that p_a and p_b most likely represent the same spine. 2) All the pixels of the spine p_a on the target shape are associated with some pixels of the dendrite backbone on the source shape, which means p_a disappears in the next time point. 3) All the pixels of the spine p_a on the target shape are associated with more than one spines on the source shape. It implies that in these regions, the spine density is high, and pixel correspondences alone cannot completely determine accurate spine associations, but provide most likely candidates for further association refinement. Therefore, local nonrigid deformation cannot guarantee the complete determination of one-to-one spine associations. Examples of the three spine association patterns are shown in Fig. 6.

VI. Spine Association Determination

In this section, 0–1 integer programming is further utilized to solve one-to-one spine associations through a maximization of a global spatial similarity metric between two dendritic structures. Let $P^k = \{p_i^k | i = 1, 2, \ldots n\}$ and $P^{k+1} = \{p_j^{k+1} | j = 1, 2, \ldots m\}$ denote spines at two time points t_k and t_{k+1}, where p_i^k denotes the ith spine at the kth time point, and n and m are the total numbers of spines at the two time points, respectively. For a spine p_i^k, we assume that m_i spines $M_i = \{(p_i^k, p_j^{k+1}) | j = 1, 2, \ldots, m_i\}$ at time point t_{k+1} are identified as the association candidates in the above local nonrigid deformation alignment step. Therefore, the total number of possible associations of all the spines at t_k is $N = \sum_{i=1}^{n} m_i$. All the possible different associations are $M = \bigcup_{i=1}^{n} M_i$. A spatial similarity metric that defines the similarity between spine p_i^k at t^k and its candidates p_j^{k+1} at t^{k+1} is calculated as follows:

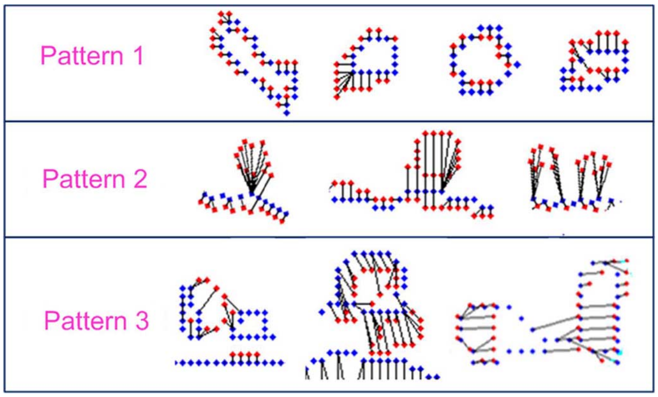

Fig. 6. Examples of three different spine association patterns.

$$\mathrm{Sim}_1\left(p_i^k, p_j^{k+1}\right) = \left[1 - \frac{\delta\left(p_i^k, p_j^{k+1}\right)}{\sum_{u=1}^{m_i} \delta\left(p_i^k, p_u^{k+1}\right)}\right] + \frac{\eta\left(p_i^k \cap p_j^{k+1}\right)}{\eta\left(p_i^k \cup p_j^{k+1}\right)}. \tag{8}$$

Here, the function δ calculates the distance between the centroids of spine p_i^k and p_j^{k+1}, and the function η computes the area of a spine. This metric suggests that for two spines, the closer the two spine centroids are, the larger overlapping area the two spines share and more likely they represent the same spine. The optimal matching strategy is to determine the optimal solution $\mathbf{z}^* = \{0/1\}^N$ that maximizes a similarity objective function [(9)] while maintaining a constraint [(10)]: each spine in either p_i^k or p_j^{k+1} can be associated with at most one spine in the other time point.

The objective function can be formalized as follows:

$$\mathbf{z}^* = \underset{\mathbf{z} \in \{0,1\}^N}{\arg\max}\,(\mathrm{f}(\mathbf{z}))$$

$$\mathrm{f}(\mathbf{z}) = \sum_{l=1}^{N} \left[\mathbf{z}(l).\mathrm{Sim}_1\left(p_{i_l}^k, p_{j_l}^{k+1}\right)\right]. \tag{9}$$

Here $\mathbf{z}$ is a $N \times 1$ vector which denotes decisions of the N possible associations. $\mathbf{z}(l) = 1$ indicates that the lth possible association is chosen, while $\mathbf{z}(l) = 0$ indicates the lth possible association is discarded. (i_l, j_l) are the spine labels in t_k and t_{k+1} in the lth possible association. Thus, the one-to-one association constraints can be formalized as

$$\mathbf{Az} \leq \mathbf{b}. \tag{10}$$

Here $\mathbf{A}$ is a $(m+n) \times N$ matrix. $m+n$ suggests that there are total $m+n$ spines in both t_k and t_{k+1} time points. Its rows indicate spines in both t_k and t_{k+1} time points, while its columns correspond to the decisions of the N possible associations. Here $\mathbf{b}$ is a $(m+n) \times 1$ vector with all 1. It means that each spine either in t_k or in t_{k+1} can be associated with at most one spine in the other time point. For the N possible associations $M_l = \{(p_{i_l}^k, p_{j_l}^{k+1}) | l = 1, 2, \ldots, N\}$

$$\mathbf{A}(k,l) = \begin{cases} 1, & ifk = i_l \\ 1, & ifk = n + j_l \quad 0 < l \leq N. \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

This above optimization problem can be solved through 0–1 integer programming. In this work, we use branch-and-bound (LPBB) based linear programming algorithm [33] to solve this optimization problem. The optimization process of LPBB is to build a searching tree by repeatedly discretizing (0 or 1) the variables (branching) and pruning the tree branches based on the optimal value of the node (bounding) computed by linear programming. Note that although applying integer programming to biological object tracking has been previously investigated in [21], we can clearly see that our algorithm outperforms the algorithm in [21] through a direct performance comparison with [21]. This is mainly due to: 1) the deformable registration process in our algorithm is capable of optimally aligning two dentritic shapes, and 2) the similarity metric introduced in our work extracts and includes more features from dendritic structures than [21].

Also, it is noteworthy to point out that features like 1) the distance between spine centroids and 2) the areas of the spines in two time points have been explored in Al-Kofahi *et al.*'s early work [1]; however, besides the two features, in our algorithm dendritic branch information implicitly guides the nonrigid local deformation (see Section V-B). And, during the procedure of linking reappearing spines (see the follow-up Section VII), neighborhood information of individual spines is utilized to determine the spine association pairs.

VII. Linking Reappearing Spines

Once spine associations between every two consecutive time points are obtained, trajectories of spine centroids are analyzed. The goal of this analysis is to eliminate false spine associations based on the spatial locations of spine centroids. For each labeled spine p_i that is observed from time point t_n to t_m, its centroid at each time point is first calculated. The mean centroid of p_i, defined as the mean of spine centroid across several time points, is denoted as $\mu_i = (\bar{x_i}, \bar{y_i})$. A 2D Gaussian $\mathrm{N}_i(\mu_i, \bar{\sigma})$ is fitted to model the spread of the spine centroids of each labeled spine. The center of each Gaussian lies at the mean centroid μ_i, and $\bar{\sigma}$ is set to be the standard deviation of the distance between μ_i and individual spine centroids c_{ij} at different time points (i denotes the spine label and j indicates which time point the spine centroid is at). Every spine centroid c_{ij} is examined by the Gaussian model N_i defined above. If $P(c_{ij} | \mu_i, \bar{\sigma})$ is less than a prespecified threshold (0.75 is used in this work), we consider that the assigned association for spine p_i at time point t_j is false. Therefore, all spines labeled as i at and after time point t_j will be assigned a new label. Associations of these spines will be reconsidered during the reappearing-spine-linking stage.

The morphology of spines changes rapidly over time. In certain circumstances, some spines even completely disappear for a certain period. Then, new spines are observed in approximately the same positions on the dendrite. Three main reasons may cause this disappearing-and-reappearing problem. 1) Spines physically disappear at a certain period and then reappear in the same place. Existing neuroscience literature [15], [25], [35] has well documented this phenomenon. 2) In this work, spines are detected in the MIP images. During the projection, some spines may be occluded by other dendritic structures. Therefore, spines may just visually disappear. 3) Spines with a low intensity are misdetected at certain time points.

Our strategy of linking reappearing spines are described as follows. First, a candidate linking list for spines that exist only within a time interval is created. Each entry in the list includes three elements: the label of the spine, the first frame where the spine appears, and the last frame where the spine exists. From this list, candidate association pairs that meet the following requirements are identified: 1) The distance between the means of the two spine centroids should be within a certain range and 2) temporal overlap does not exist between the two spines. For example, a spine observed from time point t_1 to t_4 cannot be associated with another spine existing from t_3 to t_5, since they coexist at t_3 and t_4.

Subsequently, to link reappearing spines, we further examine the spatial similarity between two spines in possible associations, and then optimal spine associations among candidate associations are determined through global similarity maximization (Section VI). In this work, the disappearing-and-reappearing linking problem is solved through graph matching. Generally, the graph matching problem [4] can be defined as follows. Given two graphs $G_i = (V_i, E_i)$ and $G_j = (V_j, E_j)$, with $|V_i| = |V_j|$, the problem is to find a one-to-one mapping $f : V_i \rightarrow V_j$ such that $(e_m, e_n) \in E_i$ if $(f(e_m), f(e_n)) \in E_j$. When such a mapping f exists, G_i is said to be isomorphic to G_j. This type of problem is called *exact graph matching*. However, in most practical cases, the numbers of vertices are different in the two graphs (called *inexact graph matching*). Various algorithms have been proposed to solve the above graph matching problems such as finding the best matching through an optimization process such as Bayesian framework [11] and EM algorithm [10], [14]. Decision trees [36] and neural networks [24], [32] have also been employed to solve graph matching.

Thus, to calculate the spatial similarity for spines p_i and p_j in a possible association, two local graphs G_i and G_j are built. Since a disappearing-and-reappearing spine may not appear in all time points, a local graph of a spine is defined in the time points that the spine appears. Taking graph G_i (the local graph of p_i) as an example, the nodes in G_i include spine p_i and its neighboring spines within a certain distance in the same branch. The local graph has one node per spine. To maintain the coherence of the two graphs, spines that exist in all time points can be considered as neighbors of other spines. Edges of the graph connect p_i to its neighbors. Each of the edge weights $\{w_{ie} | e = 1, 2, \ldots, m\}$ is defined as the distance between the mean centroids of the two spines, where m is the number of neighbors of p_i within a certain distance. The spatial similarity between graph G_i and G_j is defined as

$$\mathrm{Sim}_2(G_i, G_j) = -\left|\frac{\sum_{e=1}^{m} w_{ie} - \sum_{f=1}^{n} w_{jf}}{\sum_{e=1}^{m} w_{ie} + \sum_{f=1}^{n} w_{jf}}\right|. \tag{12}$$

Here m and n are the numbers of edges in the two graphs, respectively. This spatial similarity metric is defined based on the key observation that the relative spatial relationship between spines and their neighbors are stable. 0–1 integer programming [33] is employed again to solve spine associations through global spatial similarity maximization. The formulation of 0–1 integer programming is the same as the one used in Section VI, except here a new similarity metric Sim_2 is used.

Note that the linking approach proposed by Li *et al.* [22] shares certain similarities with our algorithm. However, the major difference between our algorithm and theirs is that, when the similarity metric is computed, our algorithm optimally utilizes the structure information of the neighbors in spatio-temporal space by constructing a local graph for each disappearing-and-reappearing spines, and then generate a similarity metric between two graphs. However, in the work of [22], Li *et al.* only use the information of individual spines for similarity metric calculation, which is not optimal and less robust.

VIII. Experimental Results and Validation

To evaluate the performance of our algorithm, 6 regions of interest (ROIs) from three time-lapse datasets were selected. Dataset #1 contains eight time points, Dataset #2 contains seven time points, and Dataset #3 contains eight time points. Spines and dendrite backbones were first detected through a curvilinear structure detector with negligible manual interventions. Then, individual spines are tracked simultaneously through eight time points. Fig. 7 shows the tracking results of ROI1 in Dataset #1. Each rectangle represents a spine, and the same spines in different time points are drawn in the same color. The reappearing spine in red circle at time point t_3 is linked with the spine at time point t_1. Fig. 8 shows the tracking results of ROI4 in Dataset #2.

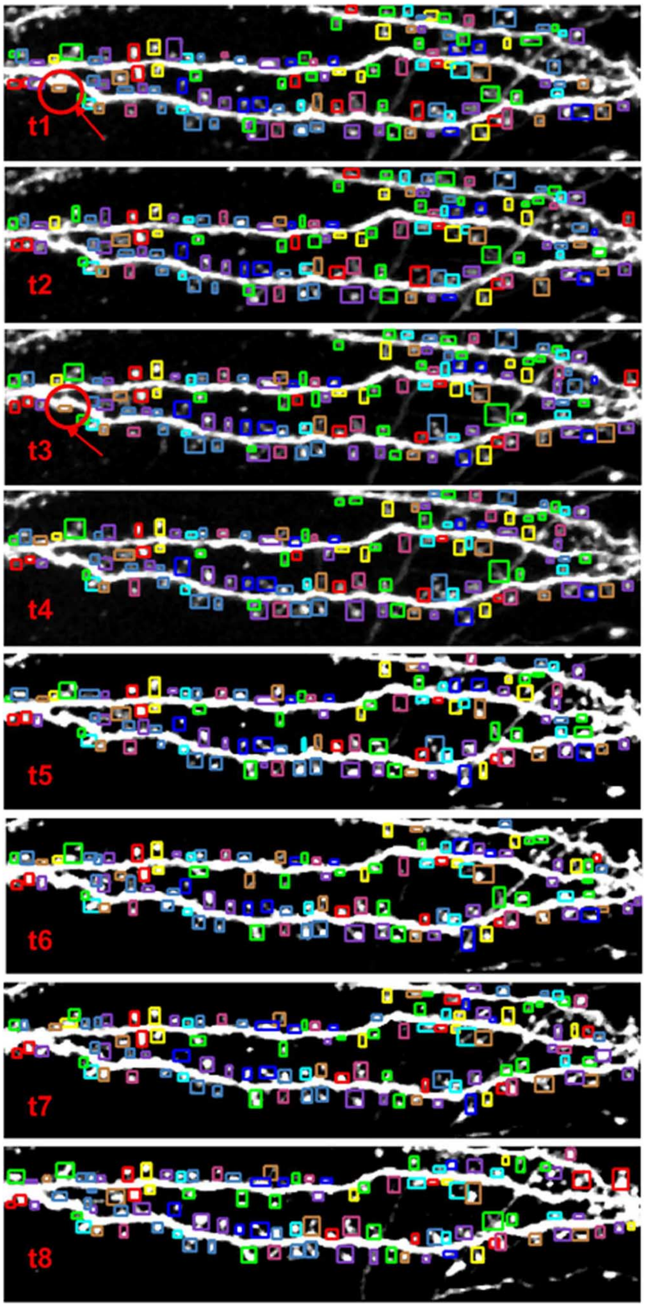

Fig. 7. Tracking results of ROI1 in Dataset 1. Each rectangle represents a spine, the same spines in different time points are drawn in the same color. The reappearing spine in red circle at t3 is linked with the spine at t1.

The results by our algorithm were validated with the ground truth in which the spines were manually labeled and tracked by a biology specialist. We also compared the performance of our tracking algorithm with a state-of-the-art cell tracking algorithm [21] and two dendritic spine tracking algorithms [13], [18]. The comparison results are shown in Figs. 9 and 10. In Fig. 10, "elimination" means how many spines disappeared in the next time point, and "formation" means how many spines are considered as new spines in the next time point.

From Fig. 9 and Table 10, we observed that our spine tracking algorithm achieved a higher accuracy than other tracking algorithms, and Li *et al.*'s method [21] and Fan *et al.*'s method [13] outperformed Koh *et al.*'s method [18]. It is not surprising, considering that Koh *et al.*'s method is just based on a heuristic rule: If the overlapping area of two spines is larger than 20% of either one of the spine areas, the two spines are assumed to represent the same one. This rule is simple and prone to fail in the region where the spine density is high. In Li *et al.*'s method [21], although the spatial information between the spine and its neighbors is incorporated into its Delaunay triangulation metric, the Delaunay triangulation is not in accordance with the dendritic structure. If two dendritic branches are too close, spines are prone to be falsely aligned. In Fan *et al.*'s method [13], dendritic structural information is incorporated into graphs, and dynamic programming is employed to maximize a posterior probability for graph mapping. However, dynamic programming is a forward optimization technique, assuming that the existing partial solution to the current decision is optimal and it would not resort to previous decisions to update current state. Although it

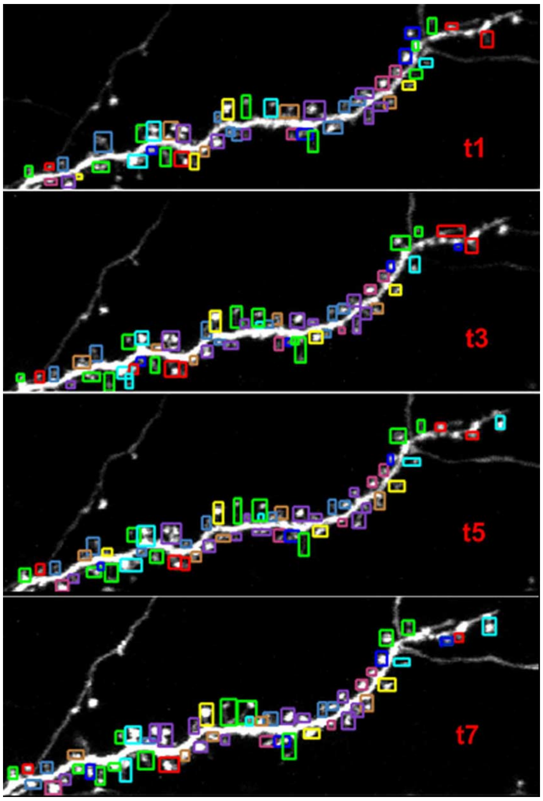

Fig. 8. Tracking results of ROI4 in Dataset 2. Each rectangle represents a spine, and the same spines in different time points are drawn in the same color.

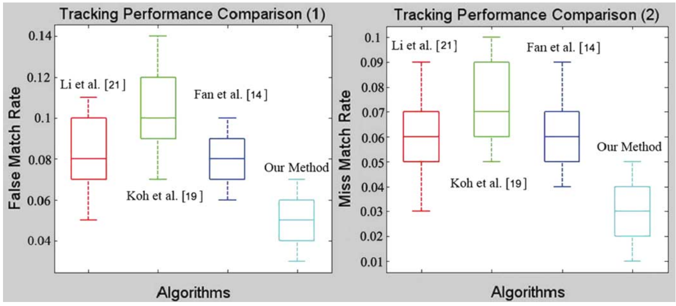

Fig. 9. Tracking performance comparison with a cell tracking method [21] and two spine tracking algorithms [13], [18] for 6 ROIs in 3 Datasets. Here, the "False Match Rate" is the ratio of the number of falsely matched spines to the total number of spines, and the "Miss Match Rate" is the ratio of the number of miss matched spines to the total number of spines.

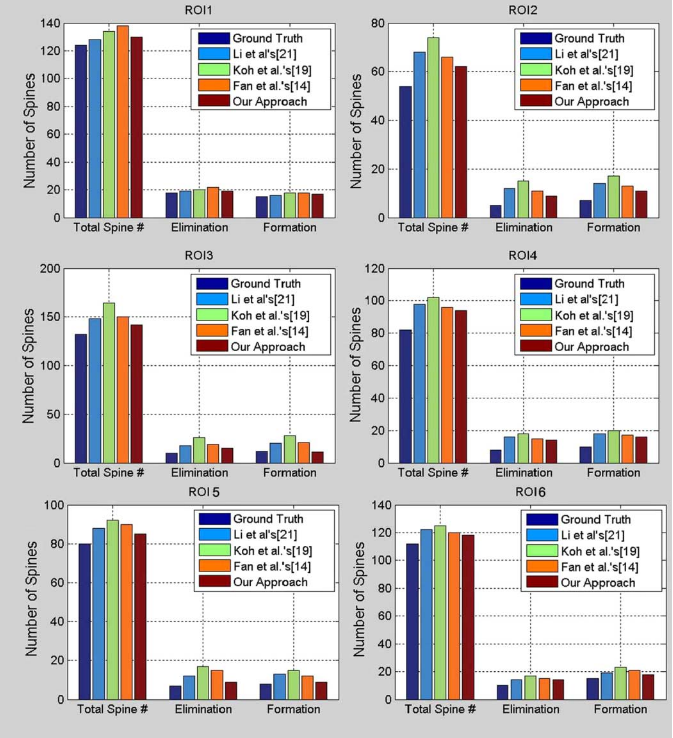

Fig. 10. Tracking performance comparison with a cell tracking method [21] and two spine tracking algorithms [13], [18]. ROI1 and ROI2 are in Dataset 1. ROI3 and ROI4 are in Dataset 2. ROI5 and ROI6 are in Dataset 3.

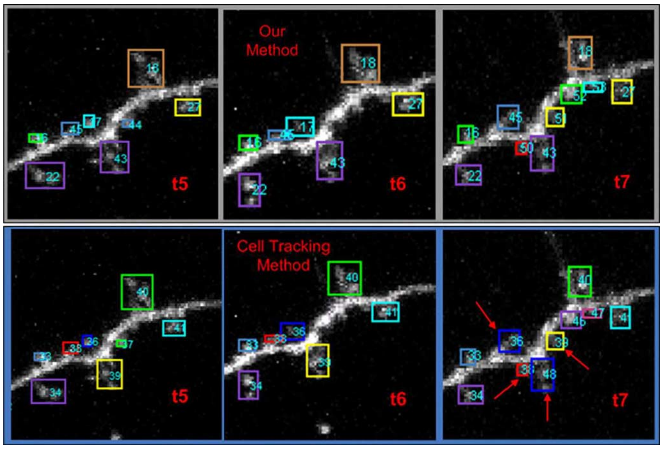

Fig. 11. Performance comparison of our method with a cell tracking method [21]. The numbers are the labels of spines. The red arrows indicate the tracking errors. This is a part region of ROI2 in Dataset 1.

is a global optimization technique, matching errors may be accumulated during the optimization process. Fig. 11 shows an example of the performance comparison between our approach and Li *et al.*'s approach. In Fig. 11, we zoom in a part of ROI2 at time points t_5 and t_7. Red arrows in the results of the cell tracking method indicate tracking errors. In Fig. 10, we plot how many spines are detected, formed, and eliminated in a time period if different methods are used.

To the best of our knowledge, our approach is the first algorithm that can handle the spine disappearing-and-reappearing problem. One such example is shown in Fig. 7. The comparison with the ground truth are shown in Table I. From Table I, we can see that our algorithm performed well in linking reappearing spines. As discussed in Section VII, in some situations, a few spines did not physically disappear; they may just be misdetected in certain time period in the spine detection step. Therefore, our algorithm can also be generalized to compensate detection errors in the bottom up tracking framework.

We implemented our algorithm and methods of comparison in [13], [18], [21] using C++ and Matlab. The average computing time for tracking spines in 7 time points is 3 min using our method, about 1.5 min using Li *et al.*'s method [21], 1.2 min using Fan *et al.*'s method [13], and 46 s using Koh *et al.*'s method [18], on a computer with a 2.20-GHz Intel Core 2 Duo CPU and 3G Memory. In general, our algorithm is measurably slower than the other three algorithms [13], [18], [21], although

TABLE I
GROUND TRUTH VALIDATION FOR LINKING REAPPEARING SPINES. ROI1 AND ROI2 ARE IN DATASET 1. ROI3 AND ROI4 ARE IN DATASET 2. ROI5 AND ROI6 ARE IN DATASET 3

ROIs	Ground Truth		Our Method	
	Total Spine	Reappeared	Reappeared	False Positive
ROI1	124	2	2	0
ROI2	54	2	1	0
ROI3	132	5	3	1
ROI4	82	2	3	1
ROI5	80	3	2	0
ROI6	112	4	5	2

our algorithm achieves a higher accuracy (Figs. 7–11). Compared with the three algorithms of comparison, computing efficiency is one major limitation of our current approach, which we plan to address in the future. In our algorithm, the FFD (Section V-B) takes the majority of the computing time.

IX. DISCUSSION AND CONCLUSION

In this paper, a novel approach for dendritic spines tracking in time-lapse microscopy neuron images is proposed. By incorporating the spine and dendrite branch association information into a local nonrigid deformation energy function, dendritic structures in different time points can be properly aligned. Then, spine associations are obtained through a global similarity maximization. After that, trajectories of spine centroids are analyzed and invalid associations are discarded. A novel strategy to link reappearing spines is also proposed.

To validate our algorithm, we compared our tracking results with the manually labeled ground truth and three state-of-the-art tracking algorithms [13], [18], [21]. Comparison results showed that our algorithm is more accurate and robust than these algorithms, and can effectively track a large number of spines simultaneously.

However, besides the aforementioned computing efficiency issue, our current algorithm has a number of other limitations. First, in our work, some dendritic information is unavoidably lost during the MIP projection. Second, in the nonrigid deformation step, the whole dendritic structure is considered as one shape. This representation may affect the locality of the nonrigid deformation. Third, our tracking algorithm is a bottom-up approach in which objects are first detected and then temporal associations are solved by maximizing a similarity function. Since in our current algorithm, spine detection and spine tracking are two separate procedures, the spine detection algorithm cannot benefit from the tracking outcomes. In the future, we plan to combine the two procedures together to further improve the detection and tracking accuracy. In addition, to provide more morphology information to the biologists, we plan to directly detect and track dendritic spines in 3D space.

REFERENCES

[1] O. Al-Kofahi, R. J. Radke, S. K. Goderie, Q. Shen, S. Temple, and B. Roysam, "Automated cell lineage construction: A rapid method to analyze clonal development established with murine neural progenitor cells," *Cell Cycle*, vol. 5, no. 3, pp. 327–335, 2006.

[2] S. Avidan, "Support vector tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2001, pp. 184–191.

[3] N. Becker, C. J. Wierenga, R. Fonseca, T. Bonhoeffer, and U. V. Nagerl, "LTD induction causes morphological changes of presynaptic boutons and reduces their contacts with spines," *Neuron*, vol. 60, no. 4, pp. 590–597, 2008.

[4] E. Bengoetxea, "Inexact graph matching using estimation of distribution algorithms," Ph.D. dissertation, Ecole Nationale Supérieure des Télécommunications, Paris, France, Dec. 2002.

[5] P. Besl and N. McKay, "A method for registration of 3D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, pp. 239–256, 1992.

[6] Y. Boykov and D. Huttenlocher, "Adaptive Bayesian recognition in tracking rigid objects," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2000, pp. 697–704.

[7] M. R. C. Veenman and E. Backer, "Resolving motion correspondence for densely moving points," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 23, no. 1, pp. 54–72, Jan. 2001.

[8] J. Cheng, X. Zhou, E. Miller, R. M. Witt, J. Zhu, B. L. Sabatini, and S. T. C. Wong, "A novel computational approach for automatic dendrite spines detection in two-photon laser scan microscopy," *J. Neurosci. Methods*, vol. 165, 2007.

[9] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of nonrigid objects using mean shift," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2000, pp. 142–149.

[10] A. D. J. Cross and E. R. Hancock, "Graph matching with a dual-step EM algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1236–1253, Nov. 1998.

[11] A. D. J. Cross, R. C. Wilson, and E. R. Hancock, "Inexact graph matching using genetic search," *Pattern Recognit.*, vol. 30, no. 6, pp. 953–970, 1997.

[12] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. G. Hill, M. O. Leach, and D. J. Hawkes, "Nonrigid registration using free-form deformations: Application to breast MR images," *IEEE Trans. Med. Imag.*, vol. 18, no. 8, pp. 712–721, Aug. 1999.

[13] J. Fan, X. Zhou, J. G. Dy, Y. Zhang, and S. T. Wong, "An automated pipeline for dendrite spine detection and tracking of 3D optical microscopy neuron images of in vivo mouse models," *Neuroinformatics*, vol. 7, no. 2, pp. 113–130, 2009.

[14] A. W. Finch, R. C. Wilson, and E. R. Hancock, "Symbolic graph matching with the EM algorithm," *Pattern Recognit.*, vol. 31, no. 11, pp. 1777–1790, 1998.

[15] M. Fischer, S. Kaech, D. Knutti, and A. Matus, "Rapid action-based plasticity in dendritic spines," *NeuronImage*, vol. 20, no. 5, pp. 847–854, 1998.

[16] X. Huang, N. Paragios, and D. N. Metaxas, "Shape registration in implicit spaces using information theory and free form deformations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 8, pp. 1303–1318, Aug. 2006.

[17] S. Knafo, L. Alonso-Nanclares, J. Gonzalez-Soriano, P. Merino-Serrais, I. Fernaud-Espinosa, I. Ferrer, and J. DeFelipe, "Widespread changes in dendritic spines in a model of Alzheimer's disease," *Cereb Cortex*, vol. 19, pp. 586–592, 2009.

[18] I. Koh, W. Lindquist, K. Zito, E. Nimchinsky, and K. Svoboda, "An image analysis algorithm for dendritic spines," *Neural Comput.*, vol. 14, pp. 1283–1310, 2002.

[19] K. Rohr, H. S. Stiehl, R. Sprengel, T. M. Buzug, J. Weese, and M. H. Kuhn, "Landmark-based elastic registration using approximating thin-plate splines," *IEEE Trans. Med. Imag.*, vol. 20, no. 6, pp. 526–534, Jun. 2001.

[20] J. Kybic, P. Thevenaz, A. Nirkko, and M. Unser, "Unwarping of unidirectionally distorted EPI images," *IEEE Trans. Med. Imag.*, vol. 19, no. 2, pp. 80–93, Feb. 2000.

[21] F. Li, X. Zhou, J. Ma, and S. Wong, "Optimal multiple nuclei tracking using integer programming for quantitative cancer cell cycle analysis," *IEEE Trans. Med. Imag.*, vol. 29, no. 1, pp. 96–105, Jan. 2009.

[22] K. Li, M. Chen, T. Kanade, E. D. Miller, L. E. Weiss, and P. G. Campbella, "Cell population tracking and lineage construction with spatiotemporal context," *Med. Image Anal.*, vol. 12, no. 5, pp. 546–566, 2008.

[23] Q. Li, X. Zhou, Z. Deng, M. Baron, M. A. Teylan, Y. Kim, and S. T. Wong, "A novel surface-based geometric approach for 3d dendritic spine detection from multi-photon excitation microscopy images," in *Proc. IEEE Int. Symp. Biomed. Imag. (ISBI)*, 2009, pp. 1255–1258.

[24] L. Y. Lyul and P. R. Hong, "A surface-based approach to 3D object recognition using a mean field annealing neural network," *Pattern Recognit.*, vol. 35, no. 2, pp. 299–316, 2002.

[25] A. Matus, "Actin-based plasticity in dendritic spines," *Science*, vol. 290, pp. 754–758, 2000.

[26] K. R. Mosaliganti, F. Janoos, X. Xu, R. Machiraju, K. Huang, and S. T. C. Wong, "Temporal matching of dendritic spines in confocal microscopy images of neuronal tissue sections," in *Proc. MICCAI Workshop Med. Image Anal. Appl. Biol.*, 2006, pp. 106–113.

[27] M. Muller, B. H. Gahwiler, L. Rietschin, and S. M. Thompson, "Reversible loss of dendritic spines and altered excitability after chronic epilepsy in hippocampal slice cultures," *Proc. Nat. Acad. Sci. USA*, vol. 90, pp. 257–261, 1993.

[28] D. R. Padfield, J. Rittscher, and B. Roysam, "Coupled minimum-cost flow cell tracking," in *Proc. Inf. Process. Med. Imag. (IPMI)*, 2009, vol. 374–385.

[29] T. Pajdla and L. Van Gool, "Matching of 3D curves using semi-differential invariants," in *Proc. 5th Int. Conf. Comput. Vis. (ICCV95)*, Washington, DC, 1995, p. 390.

[30] C. Pittenger and R. S. Duman, "Stress, depression, and neuroplasticity: A convergence of mechanisms," *Neuropsychopharmacology*, vol. 33, pp. 88–109, 2008.

[31] D. Rueckert, L. Sonoda, C. Hayes, D. Hill, M. Leach, and D. Hawkes, "Nonrigid registration using free-form deformations: Application to breast MR images," *IEEE Trans. Med. Imag.*, vol. 18, no. 8, pp. 712–721, Aug. 1999.

[32] A. Sanfeliu, F. Serratosa, and R. Alquezar, "Clustering of attributed graphs and unsupervised synthesis of function-described graphs," in *Proc. 15th Int. Conf. Pattern Recognit.*, 2000, pp. 1022–1025.

[33] A. Schrijver, *Theory of Linear and Integer Programming*. New York: Wiley, 1986.

[34] K. Shafique and M. Shah, "A non-iterative greedy algorithm for multiframe point correspondence," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2003, pp. 110–115.

[35] S. Halpain, "Actin and the agile spine: How and why do dendritic spines dance," *Trends Neurosci.*, vol. 23, pp. 141–146, 2000.

[36] K. Shearer, H. Bunke, and S. Venkatesh, "Video indexing and similarity retrieval by largest common subgraph detection using decision trees," *Pattern Recognit.*, vol. 34, no. 5, pp. 1075–1091, 2001.

[37] K. Smith, A. Carleton, and V. Lepetit, "General constraints for batch multiple-target tracking applied to large-scale videomicroscopy," in *Proc. IEEE Conf. Computer Vis. Pattern Recognit.*, 2008, pp. 142–149.

[38] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, p. 13, 2006.

[39] Y. Zhang, X. Zhou, R. M. Witt, B. L. Sabatini, D. Adjeroh, and S. T. Wong, "Dendritic spine detection using curvilinear structure detector and LDA classifier," *NeuronImage*, vol. 36, pp. 346–360, 2007.

[40] W. Zhou, H. Li, and X. Zhou, "3D neuron dendritic spine detection and dendrite reconstruction," *Int. J. Comp. Aided Eng. Tech.*, vol. 1, no. 4, pp. 516–531, 2009.