A Dynamic-Shape-Prior Guided Snake Model with Application in Visually Tracking Dense Cell Populations

Sha Yu*, Yao Lu, Derek Molloy

Abstract—This work proposes a dynamic-shape-prior guided snake model (DSP G-snake) that is designed to improve the overall stability of the point-based snake model. The dynamic shape prior is first proposed for snakes, that efficiently unifies different types of high-level priors into a new force term. To be specific, a global-topology regularity is first introduced that settles the inherent self-intersection problem with snakes. The problem that a snake's snaxels tend to unevenly distribute along the contour is also handled, leading to good parameterization. Unlike existing methods that employ learning templates or commonly enforce hard priors, the dynamic-template scheme strongly respects the deformation flexibility of the model, while retaining a decent global topology for the snake. It is verified by experiments that the proposed algorithm can effectively prevent snakes from self-crossing, or automatically untie an already self-intersected contour. In addition, the proposed model is combined with existing forces and applied to the very challenging task of tracking dense biological cell populations. The DSP G-snake model has enabled an improvement of up to 30% in tracking accuracy with respect to regular model-based approaches. Through experiments on real cellular datasets, with highly dense populations and relatively large displacements, it is confirmed that the proposed approach has enabled superior performance, in comparison to modern active-contour competitors as well as state-of-the-art cell tracking frameworks.

Index Terms—Snakes, self-intersection, cell population tracking, dynamic shape prior, global-topology regularity.

I. INTRODUCTION

S INCE the development of snakes in the seminal work by Kass *et al.* [1], they have been applied to object segmentation and widely adopted in various forms of biological image analysis. Snakes are used to define an object's outline in images under analysis by minimising an energy or cost function associated with different object properties (e.g., average intensity value), and intrinsic curve properties (e.g., smoothness). Due to the fact that the traditional internal energies of snakes encourage smooth and circular-like shapes, deformable active contours are intuitive tools for modelling cell membranes and regions. They have thus received large popularity in the cell tracking community since the 1990s. Works such as [2] and [3] are pioneering in applying the

Sha Yu and Yao Lu are with the School of Data and Computer Science, Sun Yat-Sen University, Guangzhou, China. Derek Molloy is with the School of Electronic Engineering, Dublin City University, Dublin 9, Ireland. Corresponding author: Sha Yu (email: cyan.yusha@outlook.com).

active contour model (ACM) to cell tracking. Recent works on cell detection and tracking, e.g. [4] and [5], employ the active disc model [6] and the radial snake model [7]. The ACM in [8] takes the prior of elliptical shapes. Pan *et al.* propose a new shape force that captures tubular patterns with particular types of cells in [9]. External forces that embed motion priors of cells are also considered in [5] and [7]. Nowadays, along with the intense requisite for quantitative analysis of cellular behaviors, there is a growing demand for the automated tracking and analysis of dense cell populations [10]–[13], due to the manual effort involved in marking up vast volumes of cell data for biological study. This brings forward many new challenges for the vision-based cell tracking community, and models or tools with more stable performance are required.

A. Motivation

Reliable analysis of cell migration and proliferation typically involves large numbers of cells (usually hundreds to thousands) within videos or image sequences. This makes manual tracking labor-intensive and often unfeasible [14]–[16]. Dense cell-population tracking is by itself a complicated problem, with many particular challenges, arising from the obscure boundaries of tightly packed cells, the variety of cell shapes and sizes, the varying levels of cell deformation and displacement, temporal occlusions, etc. Blurred boundaries are common in low-contrast cellular images (see Fig. 1), which means the segmentation of active contours may leak, causing under-segmentation. Snaxels on the leaked contour can be attracted by noise, or incorrect/neighboring features, resulting in segmentation error or the commonly occurring effect of snakes self intersecting, as illustrated in Fig. 2. If the snakes are region based, this self-intersection can cause a divergence of the snake, since the invalid loop changes the normal direction and thus reverses the inside/outside of the closed contour [17]. Within an edge-clutter environment, the false loops may trap irrelevant features or regions/edges of nearby cells (see Fig. 2b), leading to tracking failures. Occlusions, or cells displaying temporary intensity changes also have an influence on the snake performance, leading to undesirable convergence results. More seriously, errors from self-crossed snakes tend to have knock-on effects in the neighborhood. In tracking a small number of objects, or dense cell populations, the self-crossing effect can directly cause a center or boundary localisation error, and lead to a degraded tracking result. It is therefore important that a solution is found to solve the
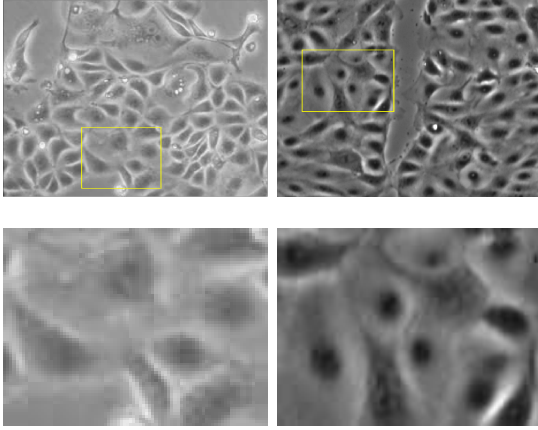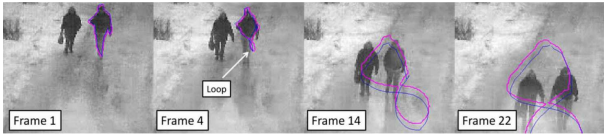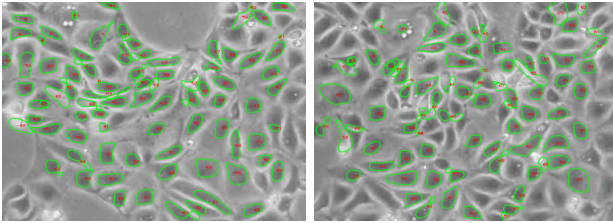
Fig. 1: Illustrating cell populations in phase-contrast image samples. Note the densely packed cells, the variety of shapes and sizes, and the ambiguous cell boundaries. Square contours denote cropped and magnified regions.

(a) Images taken from [17].

(b) From left to right, Frame# 40 and 80.

Fig. 2: Illustrating the self-intersection effect of snakes in segmentation/tracking.

self-crossing problem. Other issues that also affect a snake's performance in object segmentation and tracking are the snaxels' uneven-distribution effect and the contour shrinkage problem. However, these problems are usually overlooked or ignored.

B. Related Work

During the past decades, the snake model and its extensions have developed greatly, particularly for modelling cell morphology and movement. In spite of this, the inherent topology defect remains with parametric snakes, due to the fact that the traditional model lacks a global topology constraint. Snakes tend to develop false loops, particulary during the tracking phase due to noise, clutter, occlusion, large motion, etc. In fact, none of the aforementioned snake models are specifically targeted at the self-crossing problem. By using algorithms such as [4], [6], [8] with circular or elliptic priors, or snake models with more specific shape priors [18]–[20], the snake self-crossing problem can be side-stepped. However, these methods generally have strong shape priors that exclude them
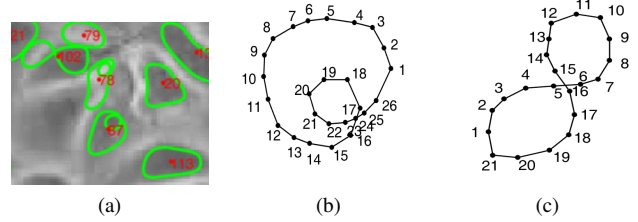
Fig. 3: Inside and outside self-crossed snakes on (a) real and (b)-(c) synthetic images.

from tracking an object with flexible deformation, or multiple objects with a variety of shapes and sizes, such as dense cell populations.

From the small amount of literature available that discusses the self-crossing problem, it is noted that the underlying problem persists, since existing solutions cannot preclude the problem from re-occurring. In [17], Nakhmani *et al.* develop an algorithm for automated self-crossing detection, after which a process for accurately locating the knot points is needed, and incorrect loops are removed or the contour is split. Another approach in [21] is based on the line-segment intersection strategy. Ji and Yan proposed a raster-inspection mechanism [22], by checking if the same raster location is plotted more than once by (densely interpolated) points on the snake. Note that, after self-crossing detection by those aforementioned methods, the snake contour is re-initialized. However, re-initialization cannot stop the re-occurrence of the self-crossing section, as the image information that caused the problem remains. Other methods for detecting snake self-crossing are referred to in [17], [22], [23]. Existing work ignores the intrinsic fact that the traditional snake model only has a local range regularity, which is not sufficient to retain a desirable global topology for the contour. As demonstrated in Fig. 3, the local regularity mechanism won't penalise the self-crossed snakes or contours, since the smoothness requirement is already satisfied. In the family of non-parameteric active contours, much effort has been made to control or reduce the topology sensitivity, i.e., stopping the contour from undesirable splitting or merging [24]–[26]. Existing works such as [16] and [27] employ topology constrained level-sets to multiple cell tracking. However, the strategies in the non-parametric family cannot be borrowed, since the physical configurations of the two families are quite different. Still, other works can be referred to in [28]–[31], where active contour/shape models have been applied to segmenting different modes of medical images and a wide range of targets.

Another issue also weakening the model's performance is the uneven-distribution effect of the snaxels, especially when edge signals have uneven intensity or contrast. Unevenly converged snaxels affect the segmentation result in the current frame and also degrade the subsequent tracking. However, this problem is also overlooked and there is limited literature on the topic. In [32], the original L1-norm term that constrains the contour length is replaced with a squared L2-norm based regularity, in the B-spline active contour framework. This

helps to enforce an equidistant spacing for the spline's control points. In the case of point-based snakes, the quadratic L2-norm based regularity is used more commonly; however, it encourages short contours.

In order to solve either the self-crossing problem, or the improper distribution or aggregation of snaxels, one may resort to existing shape-priors-based regularities. However, for tracking an object with flexible deformation, or segmenting dense cell populations with various modes of shape and size, restricting snakes with relatively strong shape constraints is inappropriate. In this paper, a novel dynamic shape prior is proposed for the parametric snake model. In this way, we aim to significantly enhance the model stability while also respecting the deformation flexibility of snakes.

II. THE DYNAMIC-SHAPE-PRIOR GUIDED SNAKE MODEL

A traditional parametric ACM or snake, contains internal, external and constraint energy terms [1]. The snake can be represented by a closed curve $C(s) = (x(s), y(s))$, that is parameterised by $s \in [0, 1]$. In practice, the snake contour is usually shaped by a number of discrete control points, also called snaxels. The traditional snake works by minimising an associated energy functional $E(C)$, as defined below:

$$E(C) = \int_0^1 (\alpha|C_s(s)|^2 + \beta|C_{ss}(s)|^2 + \\ \gamma E^{img}(C(s)) + \kappa E^{con}(C(s)))ds, \quad (1)$$

where the first two terms respectively constrain the elasticity and smoothness of the contour. E^{img} represents the external energy term, encouraging the snaxels to converge onto desired features in the image. E^{con} gives rise to external constraint forces. α, β, γ, κ are the associated weights. The snake that minimises $E(C)$ must satisfy the Euler equation:

$$-\alpha C_{ss} + \beta C_{ssss} - \gamma F^{img}(C) - \kappa F^{con}(C) = 0 \quad (2)$$

where the image force $F^{img}(C) = \nabla E^{img}(C)$, and $F^{con}(C) = \nabla E^{con}(C)$. The snake will move because of the competition between the forces, and will reach equilibrium when the forces are balanced by each other.

A. Topology and Distribution Constraints based Guiding Force

In order to deal with the self-looping problem, existing works try to seek efficient ways to check the order of the snaxels, such as [17] and [33]. After that, processes are required to delete in particular the knot snaxels or entirely remove incorrect loops, and then to re-initialize the snake. Since the order of the snaxels can be badly affected with multiple knots and incorrect loops on the snake, automated order-checking or knot-localisation is typically non-trivial. More importantly, the self-intersection may keep re-occurring as the underlying problem remains. Inspired by these observations, we suggest to initiatively supply the snaxels a correct order, and assign preferred seats to the snaxels along the contour. The proposed method first extracts a binary mask according to the run-time shape of the snake. As illustrated in Figs. 4a, and 4d,

a so-called minimum-envelop (ME) based template is first introduced, which is simply the boundary contour that closely wraps around the snake region. Other forms of dynamic templates are also designed (to be explained respectively in section II-B). Based on the dynamic template, a new constraint force is proposed, called the deformation-guiding force F^{dg}, with an initial formulation defined as,

$$F^{dg,0}(C(s), \tilde{C}(s)) = \tilde{C}(s) - C(s) \quad (3)$$

where $\tilde{C}$ represents the template contour. $\tilde{C}(s)$ gives the particular guiding point for the snaxel $C(s)$. So, the attraction force $F^{dg,0}$ aims to pull each snaxel $C(s)$ closer to the position $\tilde{C}(s)$. Please note that the template $\tilde{C}$ is always a simple and continuous curve, which involves no self-loops. So, driven by the guiding force, the snake automatically pursues a desired global topology. In order to find an optimal guiding point for each of the snaxels, $\tilde{C}$ can be obtained by contour registration, namely solving the equation below,

$$\underset{\tilde{C}}{\operatorname{argmin}} \int_0^1 |\tilde{C}(s) - C(s)| ds, \quad (4)$$

with $\tilde{C}$ parameterising the ME boundary of the snake-covered region. In fact, given the same ME mask, the boundary curve has no unique parametrisation, i.e., setting the starting point to a different position (along the boundary) leads to a different parametrisation. To facilitate implementation, the template contour $\tilde{C}$ is also discretely represented as a number of evenly sampled control points (referring to number of the snaxels). See the red control points in Fig. 4b, and 4e. So, Eqn. (4) can be trivially solved by plugging every possible formulation of $\tilde{C}$, namely varying the coordinates of $\tilde{C}(s = 0)$ along the ME boundary. This essentially performs a (discrete) contour alignment, where each snaxel at $C(s)$ is thus associated to a guiding point at $\tilde{C}(s)$, as illustrated in Figs. 4c, and 4f.

Thanks to the guiding-force mechanism, the uneven-distribution effect of the snaxels is also handled with ease, since the guiding points are evenly sampled. As demonstrated in Figs. 4b, and 4e, the snaxels not only follow a correct order but also achieve an equidistant distribution. Compared with the classic continuity term that gradually leads to a longer contour, the even-distribution (ED) constraint helps the snake to maintain continuity while also controlling the contour length.

Before presenting the final formulation, an important fact needs to be explained: if the snaxels are moved directly towards or onto the paired positions, an undesired effect is that the contour will contract from iteration to iteration. See the first two graphs in Fig. 5. Noting that the contours are only driven by the guiding force, the contraction is thus not caused by the (internal) elastic force, but relates with the guiding direction of the attraction force. To accommodate the contraction issue, a force-projection (FP) process can be considered (to be explicitly defined in Eqn. (5)). The idea is to guide the snaxel closer to the paired position along the local tangent direction. In the third graph in Fig. 5, the snaxels are translocated along the tangent directions, also approaching the guiding points, while not altering the size of the contour region.
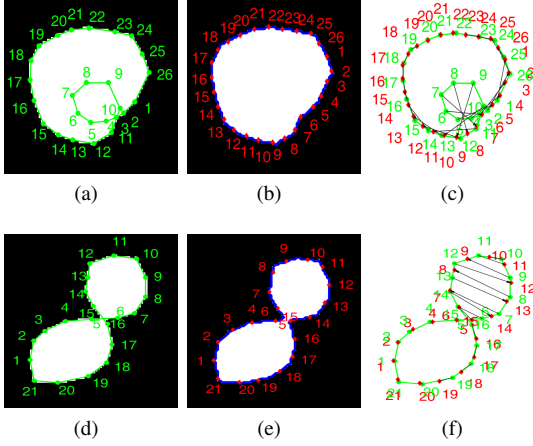
Fig. 4: Within each row: First, the ME based template is extracted from the snake (denoted with a green colour). Then, the boundary contour (with blue colour) of the ME template is discretely sampled to get a sequence of control points (red ones). The third column shows that the sampled control points are paired with the snaxels by contour alignment. The arrows connect the snaxels with the associated guiding points.
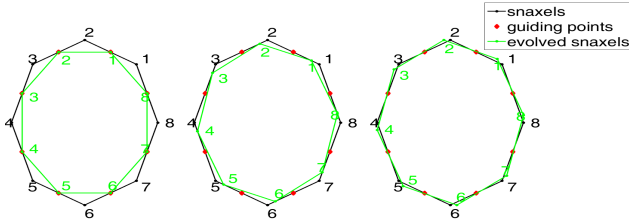

Fig. 5: Left: Directly moving the snaxels onto the guiding points, where a contour-shrinkage effect is observed, by comparing the region sizes of the evolved (green) contour and the original (black) contour. Middle: Gradually moving the snaxels towards the guiding points. See the green contour also gets a reduced area. Right: The snaxels are translocated along local tangent directions, also approaching the guiding points, while the size of the contour region is retained.

However, there is another issue to be discussed. For a snake free of self-looping, it is acceptable to universally apply the FP process. Gradually translocating the snaxels along tangent directions ends up with a sequence of evenly-spaced snaxels. While for a snake already with a self-loop, e.g. in Fig. 3c the order of the 6th to 15th snaxels is corrupted. These snaxels must swap positions, or the entire incorrect loop must be removed. Actually, it is unclear which approach is better or safer. Thankfully, moving these snaxels straight towards the registered points, can not only achieve the position re-ordering, but also gradually shortens the invalid loop. Thus, it is rational to have a divide-and-conquer strategy, by classifying the snaxels into the FP-process group or not. Here, an efficient method is suggested by checking the intensity values along the normal vector. As illustrated in Fig. 6, a snaxel on a valid loop satisfies the simple rule: the intensities (in the binary mask) change from bright to dark from the inward normal towards

the outward normal. If that condition is violated then there is a clear indication of a self-crossing. Correspondingly, the final formulation of the guiding force is defined as,

$$F^{dg}(C, \tilde{C}) = (1 - \Omega)(\tilde{C} - C) + \Omega(<\tilde{C} - C, N^\perp > N^\perp) \quad (5)$$

where Ω encodes the divide-and-conquer strategy, to be specified below. Without ambiguity here, the curve parameter s is omitted. Recall that $(\tilde{C} - C)$ describes the same attraction force as in Eqn. (3). $< \cdot, \cdot >$ stands for the inner product operator and $N^\perp(\cdot)$ denotes tangent vector. So, $< \tilde{C} - C, N^\perp >$ measures the magnitude of the attraction force that is projected on the tangent direction, and $N^\perp$ specifies the force direction. For symbol consistency, we employ $N(\cdot)$ and $-N(\cdot)$ to represent outward normal and inward normals respectively. Explicitly, $\Omega(s)$ is defined as,

$$\Omega(s) = \begin{cases} 1 & \text{if } \mathcal{M}_{\tilde{C}}(C(s) - N(s)) > \mathcal{M}_{\tilde{C}}(C(s) + N(s)), \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

So, for snaxels with $\Omega(s) = 1$, the FP process is applied; for $\Omega(s) = 0$, the original attraction force is employed. $\mathcal{M}_{\tilde{C}}$ denotes the binary template mask, as shown in Fig. 6, with only the elements in the template region set to one. $C(s) + \nu N(s)$ denotes the position away from $C(s)$ by 1 unit length along the outward or inward normal, respectively for $\nu = 1, -1$.

B. Potential Types of Dynamic Templates

Except for the ME template, two other types of dynamic templates are considered: major-blob (MB) and convex-hull (CH) based, as illustrated in Fig. 6. The MB and CH based templates are constructed as follows. For the MB based template, the original binary mask (covered by the snake) is first eroded, so potential blobs from different loops are divided. Then, the biggest blob is reconstructed by image dilation, and the smaller blobs are removed. Fundamentally, the ME based shape prior assumes valid and invalid loops both contain useful information, more or less about the final shape and position of the target. So, all the snaxels are taken into account in the construction of the template; in fact the MB type prior excludes invalid loops in the template construction, with an assumption that invalid loops have a relatively smaller area (at least at the beginning of the self-intersection); The CH type prior assumes convex-shape targets.

Similar to the ME based case, in MB and CH based ones, the snaxels can be efficiently classified into FP- and non-FP-process groups, according to the binary values pointed by the inward and outward normal vectors. Also, see illustrations in Fig. 6b and 6c.

C. Implementation and Evaluation

1) Implementation: The implementation of the DSP G-snake model is straightforward. It involves finding the steady-state solution of the equation below:

$$\frac{C^{t+1} - C^t}{\Delta t} = -\alpha C^t_{ss} + \beta C^t_{ssss} - F^{img}(C^t) - \kappa F^{dg}(C^t, \tilde{C}^t) \quad (7)$$
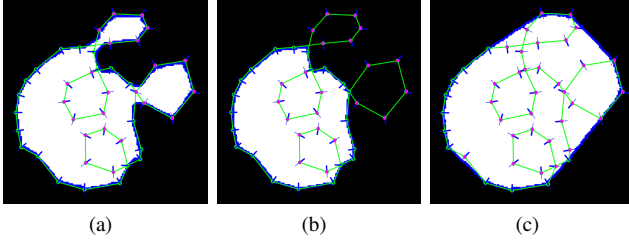
Fig. 6: (a)-(c) respectively illustrates the ME-, MB- and CH-based template masks, for the same snake. The snake is still represented as a green contour with snaxels. The blue contours are the corresponding template boundaries. The local normal vectors are represented as blue pins, with the inward and outward vectors represented respectively with thick and thin ends. According to the binary values along the normal directions, the snaxels are classified into two groups. The snaxels that do not need the FP mechanism are highlighted with magenta. These snaxels should directly follow the attraction-force mechanism, so as to eliminate invalid-loops, or achieve the position swapping (for some magenta snaxels in (a)), or to remove concavities (for some magenta snaxels in (c)).

where the superscripts denote the time index and Δt is the step size. Note, in each iteration of the snake deformation, the dynamic template $\tilde{C}^t$ is updated according to the current snake contour C^t. To summarise the steps:

- Step 0: Initialize $C^{t=0}$;
- Step 1: Extract a run-time template mask $\mathcal{M}_{C^t}$ from the current snake C^t. Template shapes include ME, MB, and CH types. If the foreground area of $\mathcal{M}_{C^t}$ is less than a pre-defined size τ_a, a circular template or a dilated template is constructed;
- Step 2: Along the boundary contour $\tilde{C}^t$ of the template mask, extract a sequence of control points with equal intervals;
- Step 3: Align the discretely sampled C^t and $\tilde{C}^t$, and associate each snaxel $C^t(s)$ with the guiding point $\tilde{C}^t(\tilde{s})$;
- Step 4: Update C^{t+1} according to Eqn. (7);
- Step 5: Stop if the snake has converged; Otherwise, update $\tilde{C}^{t+1}$, and repeat steps 1-5.

Note that a snake may contract to a small dot or an overly thin structure, which can be caused by occlusion, object exits from the image frame, and other factors. If during the snake evolution, the contour-covered region becomes too small (i.e. less than a tolerable size τ_a), we consider that a shrinkage effect has happened. In that case, a dilated template or a circular template (specified by τ_a) will be used, rather than constructing a template directly from the snake.

In the following section, the DSP G-snake model is tested on synthetic contours with varying levels of self-intersection. The weighting parameters for the snake model are set as $\alpha = 0$, $\beta = 0$, $\gamma = 0$, $\kappa = 1$. By that means, only the deformation-guiding (DG) force is activated for validating the performance of the proposed model. The convergence condition is defined

according to two conditions: First, the standard deviation of the snaxels' intervals should be sufficiently small. Secondly, the ω values (as defined in Eqn. (6)) along the contour should be all positive, i.e., invalid loops are entirely eliminated.

2) Automatically Untying Self-Crossed Contours: The ME based G-snake model is applied to synthetic contours, as shown in Fig. 7. The evolution processes of the snakes *with the FP mechanism* are illustrated from the 1st column to the 5th, with the converged contours in the 5th column. The self-crossed contours are successfully untied, no matter with simple or complex self-intersections. Note that the invalid loops are eliminated and the snaxels gradually attain equal intervals.

It is worthy of discussion that, comparing Fig. 7b with Fig. 4d, 4e, and 4f, why the upper blob disappears in Fig. 7b. The reason is that the self-untying process should not be conducted within one iteration. First, this is to avoid the undesirable effect that using the guiding-force mechanism fully cancels the work/effort of other snake forces. Also, it is important to realise that, self-looping usually happens in the case of a contour being distracted by noise, or incorrect neighboring features. That means fully conserving the self-loop region could be dangerous. The question relies on the fact that the amount of region to preserve is unknown. A conservative way is to temporally preserve the region but allow a gradual deflation, just like employing the ME G-snake. A more radical way is to totally remove the self-loop region, namely using the MB template based scheme.

For a close comparison, the last column in Fig. 7 presents the results using the ME G-snake model *without the FP mechanism*. As demonstrated, the G-snakes without the FP mechanism suffer from the impact of contour contraction. While for the case of G-snakes with FP, allowing that the invalid loops are deflated, the converged structures respect the initial shapes very well.

3) Experiments with Different Dynamic Templates: Qualitative results of applying G-snakes with MB and CH based dynamic templates are presented in Fig. 8.

To confirm the integrity of this approach, the DSP G-snake model was also tested on real images for evaluating the segmentation performance. Corresponding results are included as supplementary materials.

To summarise, the ME and MB based DSPs are both designed for general application, but with differences on how they treat invalid loops on a self-crossed contour. The CH-based shape prior is relatively application specific. Nevertheless, the three dynamic templates can be used in a flexible way in combination with practical situations.

III. DSP G-SNAKES BASED CELL TRACKING

The proposed snake model is applied to tracking dense cell populations over phase-contrast datasets. Since this is a quite challenging problem, it provides a good scenario in which to validate whether the model stability and performance is enhanced by engaging the DSP mechanism.

A. Repulsive, G-Snakes with DGVF force

The proposed cell tracker model is based on the DSP G-snake model, in combination with two other forces, namely

(a) Case I: ME wFP Iter#1,8,15,22,32; ME w/oFP Iter#27

(b) Case II: ME wFP Iter#1,7,14,31,61; ME w/oFP Iter#200

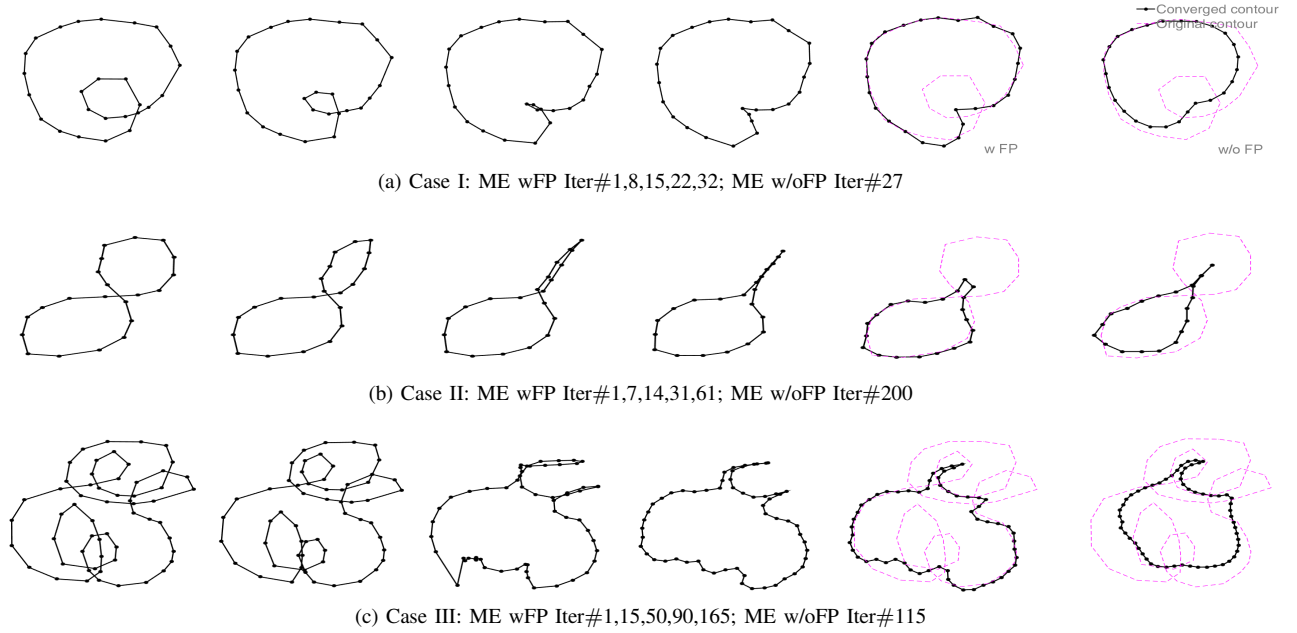(c) Case III: ME wFP Iter#1,15,50,90,165; ME w/oFP Iter#115

Fig. 7: From column 1 to 5, the self-crossed contours are automatically untied by the ME G-snake model with the FP mechanism. The fifth column includes both the initial contours (dashed contours with magenta) and the converged contour. For a close comparison, the last column presents the results from the ME G-snake without the FP mechanism (the intermediate results are omitted).
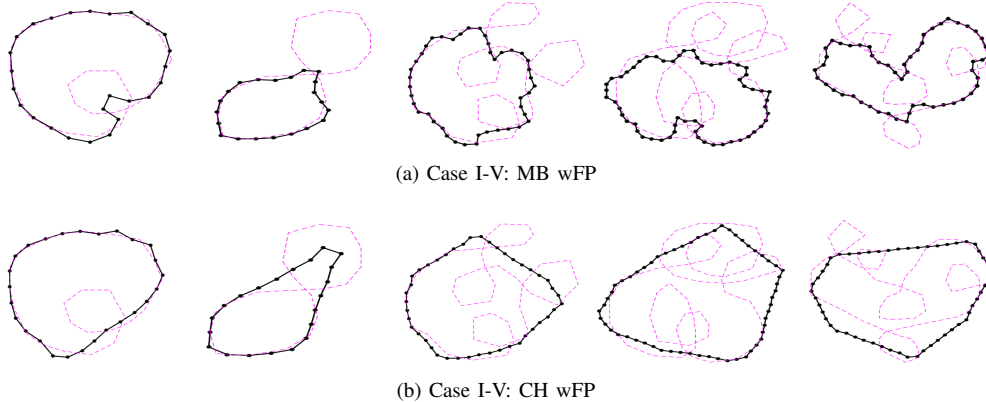

(a) Case I-V: MB wFP

(b) Case I-V: CH wFP

Fig. 8: The first row display converged results from MB based G-snakes, and the second row provides the results from CH based G-snakes, both with the FP mechanism.

an inter-object repulsion force and a dynamic gradient-vector-flow (DGVF) based image force. To be specific, the associated Euler equation can be written as below:

$$-\alpha C_{ss} + \beta C_{ssss} - \kappa F^{dg}(C, \tilde{C}) - \gamma_1 F^{dgvf}(C) - \gamma_2 F^{rep}(C) = 0 \quad (8)$$

- $F^{dg}(C, \tilde{C})$ is defined in equations (5) and (6). Only the template type of $\tilde{C}$ is to be specified from the possible types $\{ME, MB, CH, R(\tau_a)\}$. $R(\tau_a)$ denotes a small circular template with a predefined size τ_a, applied when the snake-covered region become inappropriately small.
- The same repulsive force F^r is borrowed from our recent work [34], with the specified formula as

$$F^{rep}(C(s)) = -N(s) \cdot \mathcal{M}_{-C}(C(s)) \quad (9)$$

where $\mathcal{M}_{-C}$ is a binary map, denoting the regions of all the snakes except the snake C. The repulsive force direction is $-N(s)$ (the inward normal vector), with a magnitude of 1. This gives the local repulsion force a privilege to overcome the influence of other forces, and to stop the snakes from overlapping.

- The DGVF force is defined according to the original work of Cheng *et al*, which is able to provide efficient segmentation in a clutter environment. To avoid overloading the paper, we determine not to repeat the equations (10-13) that are detailed in [35].

Note that the magnitude of the repulsion force is always 1, and the DGVF and DG force fields are both normalised. So, only three out of the five weights in Eqn. (8) need to be tuned in practice. According to the Euler equation in Eqn. (8),

the algorithm can be trivially implemented with the gradient descent method.

1) System Overview: Since cell-population tracking is a relatively complicated problem, this work also develops a full cell tracking system, that consists of the following function blocks:

- Image pre-processing, by applying the median filter to each frame in the image sequence.
- Snakes initialization: the cell detection method as described in [12] is employed, which results in a binary map with candidate cell regions (see demonstration in Fig.2 (h) in [12]). Snakes are then initialized as small circular contours that are centered at the candidate regions. Since low-level image segmentation cannot avoid the under- or over-segmentation problem, manual corrections are also involved with limited effort.
- DSP G-snake based cell trackers: With the exception of the first frame, the cell trackers are automatically initialized by the converged snakes from the last frame. They then track cells across subsequent frames. Note that the guiding-force mechanism is involved during the whole evolution process of the G-snake model. At each iteration, a new dynamic template is constructed for the G-snake, and guiding points are updated by extracting equidistant samples along the template boundary. The guiding force then works together with other types of snake forces, by following the gradient-descent method. Details are provided in Section II-C1.
- Repulsive force, and DGVF force calculation.
- Cell-Division detection: After the snakes converge in each frame, the number of cell candidates inside each snake region is checked using the same cell-detection method as described above. A positive mitosis event is determined if more than one cell candidate appears in the same snake region for three consecutive frames. This aims to lower the chance of over-segmented regions being falsely reported as mitosis events. The snake is then split and new identities are assigned. Since cell-division detection is not the current focus, only a basic strategy is employed. Thus, this cannot guarantee that all the mitosis events are captured.
- Dealing with exiting cells: For a snake whose centroid is very close to or exceed the image border, e.g., within 3 pixels, the snake will be killed.
- Measurements output, including the estimated cell regions and centroid in each frame.

The basic functional blocks are similar to our previous work [34], while the current system employs the proposed G-snake model as the core algorithm, and two new blocks are introduced for dealing with cell division and exiting events respectively. Still, note that the final tracking system is of quite simple design, with no sophisticated steps or post-processing requirements.

IV. EVALUATION AND DISCUSSION

A. Dataset Description

The performance of the developed tracking algorithm was tested on three different datasets: a dataset of wound-healing

TABLE I: Cellular datasets in the experiments

Dataset type	#of frames/ sequence	Time interval (min/frame)	#of cells	#of leaving cells	#of cells into account
WH-PEC-d1	100	4	74	1	73
WH-PEC-d5	20	20	74	1	73
MDCK1	80	10-20	82	4	78
MDCK2	100	10-20	102	14	88

pig epithelial (WH-PEC) cells, as described by [36], and two image sequences of Madin Darby Canine Kidney Epithelial (MDCK) cells (similar to [12]), all acquired using a phase-contrast microscope. The WH-PEC dataset consists of 100 frames of dimensions 300x300 ($40\mu m$x$40\mu m$). The two sequences of the MDCK data are respectively referred to as MDCK1 and MDCK2. More details about the cellular datasets are listed in Table I. WH-PEC-d1 denotes the original sequence with the time interval 4 min/frame, and the WH-PEC-d5 sequence is obtained by taking one out of five frames from WH-PEC-d1, so the time interval is 20 min/frame.

B. Parameter Setting

As explained in the last section, only three weights need to be adapted in Eqn. (8), namely α, β, κ. In the experiments, $\alpha = 0.1$, $\beta = 1$, $\gamma_1 = 1$, $\gamma_2 = 1$ are used in all the experiments. A relatively small value for α permits a weak constraint on the elastic force. Other parameters are intuitively determined as following: Each snake is discretely sampled as $sN = 40$ snaxels; The initial radius of the active contour is set as $R = 6$ for MDCK1, and $R = 7$ for the other datasets; The SR constraint is invoked when the size of the snake region is less than the threshold $\tau_a = 10$. For estimating the DGVF-based image force, the diffusion weight and the iteration number are also tuned once and fixed for all the cellular datasets. All of the involved parameters are kept exactly the same in all of the experiments, except that κ is varied in the sensitivity-analysis section.

C. Testing Results

On one hand, the experiments are conducted to evaluate the G-snake model with different types of dynamic templates. On the other hand, by replacing the G-snake with the traditional model in the proposed system, the performance of the two model types are compared. For the traditional model based tracking, within one set of tests, the snakes are not re-initialized (except after cell-division events). For the second set of experiments, the snakes are re-initialized in every frame. Besides, all the G-snake based trackers are only initialized to trigger the tracking process, and then they are left to automatically deform frame by frame.

Although the dynamic template needs to be extracted at each iteration (during the contour evolution), and so a seemly implicit re-parametrisation is involved. However, the proposed G-snake differs fundamentally to a regular snake re-sampled at each iteration (RAEI). The latter will quickly give rise to segmentation and tracking errors. To demonstrate this factor,

detailed experiments are performed in the supplementary Appendix A. It is actually dangerous to directly alter the snaxels' positions after each iteration or even too frequently (during the contour evolution). By applying the RAEI strategy, the required global-topology constraint is essentially decoupled from the other forces of the snake. This can to a large extent cancel out the work of other forces at each iteration. While, the proposed G-snake aims to couple all the forces and extra constraints in a unified formula, which enables a collaboration/competition environment.

Since ground truth data of cell segmentation is not available, only qualitative results are presented in Fig. 9. Note that $\kappa = 1$ is fixed for testing G-snakes based trackers on all the datasets. For testing traditional snakes with or without re-initialization, the exact same system is employed except for shutting down the guiding-force mechanism, i.e., setting $\kappa = 0$ in Eqn. (8).

To numerically quantify the accuracy of the proposed approach, the automated tracking results are compared against the manually annotated data that is established by a human expert. The expert was provided with a graphical user interface that allowed her to place markers at the pixel locations closest to the centers of the cells. Each marked cell in the first frame is automatically assigned a unique identity by the user interface. The expert is allowed to scroll through the slices and follow the centroid movement of the cell until the last frame. The tracking accuracy is given by the number of correct (automatically) tracked cells that are identified by the proposed approach with respect to the total number of cells that have been (automatically) taken into account in the first image. Only cells that are always in the image field over the entire image sequence are taken into account. So, earlier exiting cells are excluded. To be more precise, in the last frame of each sequence, the automated cell centroid is compared with the manually marked positions. If the deviation is within the cell radius, the tracking is considered correct. In addition, during the early stage of cell division, a snake may wrap around both of the child cells for several frames. Once the correct child cells are captured, the tracking is still considered valid for a temporary period.

In Table II, the abbreviation "re-init" stands for re-initialization. In Table III, e.g., 1185(70) means that the total number of the self-crossing events is 1185, which happens on the same 70 snakes (with repeated self-crossing). The self-crossing event (SCE) is only examined once per frame for each snake. To be specific, only for contours with area sizes > 10, the SCE is checked according to the line-segment intersection strategy. For the traditional ACM based experiments, re-initialization is processed as following: First, each snake is examined for whether a SCE is evolved; For snakes without self-crossing, a new sequence of control points are evenly sampled along the snake contour; For a snake with self-crossing, the current centroid is extracted and then the snake is re-initialized at the position as a circular contour with radius R.

In the traditional snakes-based experiments, relatively lower tracking accuracies are reported in Table II. After checking the identities of the lost tracks and the self-looped snakes, it is found that about 70% to 80% of the identities are coincident.

It is thus proved that many failure tracks are directly caused by the contour self-intersection. Particularly, blurred boundaries, close contact between neighbours, cell division and death events, and large/strong motions are major inducement factors for the SCEs. For traditional snakes, it is in fact very hard to maintain a relatively stable tracking performance without frequent re-initialization. By contrast, the G-snakes based segmentation and tracking results are not degraded over time. See a demonstration in Fig. 9. Thanks to the dynamic template based guiding-force mechanism, the global structure of the G-snakes are well preserved during the entire tracking process. The SCEs are nearly all eliminated using the proposed snake model, as reflected in Fig. 9 and Table III. A small number of cells are under-segmented due to the cell division or death events. These events are usually accompanied with appearance and intensity changes, and may continue for 3-5 frames (and also repeat). As a result, they impose large difficulties in accurate segmentation and continuous tracking. Still, one or two cells (out of the about 80 cells) get lost due to large displacement or strong deformation. This is because the existing snake models are still limited by the capture range problem. However, the problem is currently beyond the focus of this research paper. In spite of the various difficulties, the proposed snake model has greatly enhanced the accuracy of tracking. And, the rate of improvement is ranging from $10\% - 30\%$ (see Table II).

The ME, MB and CH based G-snakes have all enabled high tracking accuracies. The ME and MB based developments are both general approaches, so they can be straightforwardly applied to other scenarios of segmentation and tracking. From observation, since the SCEs are largely suppressed in the experiment, the performance difference of the two template types is not obvious. The CH based G-snakes achieved slightly higher accuracies due to an extra prior involved, that encourages the contour to capture convex shapes. This is beneficial in the tracking of dense cell populations. Since cell boundaries are usually blurred or incomplete, and the intensity or edge signals might be not constant between consecutive frames, some of the snaxels can easily get trapped in local minima. So, the convexity prior provides a soft constraint to prevent the contour structure from collapsing. This is demonstrated in Fig. 10.

Recall the definition of $\omega(s)$ in Eqn. (6). By checking if any of the $\omega(s)$ values becomes zero, a simple way is suggested to identify whether a self-crossing has occurred. Based on that knowledge, within one set of experiments, the guiding force is activated only when the contour needs to be automatically untied. Otherwise, the whole guiding-force mechanism is blocked (also including the even-distribution constraint). See Table IV, the tracking accuracies on the three datasets (with relatively low frame rates) are slightly decreased, without always running the ED constraint. The statistics are compared with the first column of Table II).

D. Parameter Sensitivity Analysis

This section aims to evaluate the parameter sensitivity for κ, varied in the set $\{0.2, 0.4, 0.6, 0.8, 1, 1.2\}$. All of the

(a) WH-PEC-d5, Frame# 1(1), 5(21), 13(61), 17(81), 20(96).

(b) MDCK1, Frame# 1, 20, 40, 60, 70, 80.
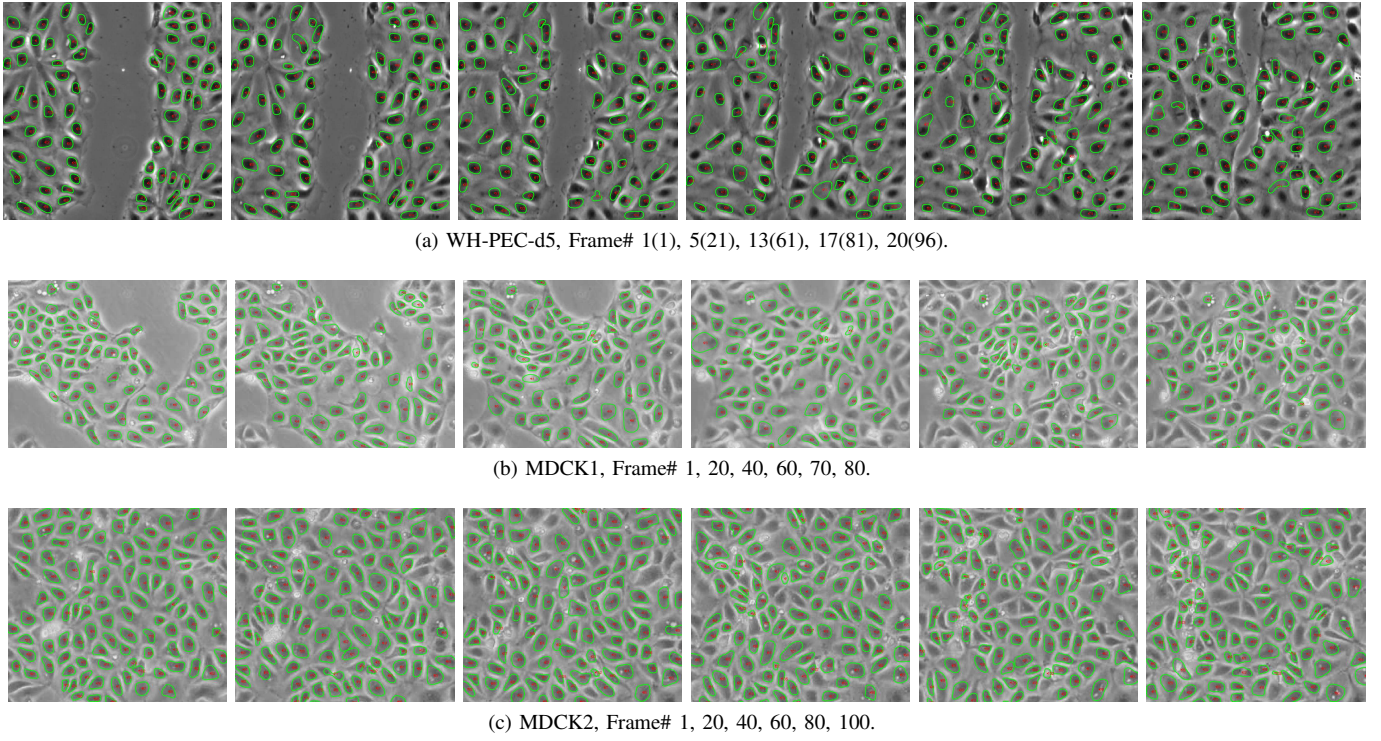

(c) MDCK2, Frame# 1, 20, 40, 60, 80, 100.

Fig. 9: Illustrating the cell segmentation and tracking effects by using the proposed ME G-snake based system on different image sequences. In (a) e.g. with Frame# 5(21), 5 denotes the frame number in WH-PEC-d5, and 21 indicates the frame number in the original data WH-PEC-d1.

TABLE II: Quantitative results for tracking accuracies

	G-snake			Regular snake	
	ME	MB	CH	w/o re-init	w re-init
WH-PEC-d1	(69/73)94.52%	(70/73)95.89%	(71/73)97.26%	(63/73)86.30%	(69/73)94.52%
WH-PEC-d5	(69/73)94.52%	(69/73)94.52%	(69/73)94.52%	(53/73)72.60%	(64/73)87.67%
MDCK1	(75/78)96.15%	(75/78)96.15%	(75/78)96.15%	(49/78)62.82%	(60/78)76.92%
MDCK2	(85/88)96.59%	(83/88)94.32%	(86/88)97.73%	(65/88)73.86%	(76/88)86.36%

(a) ME G-snake #19, frames #2-11.

(b) CH G-snake #19, frames #2-11.

(c) ME G-snake #37, frames #30-42.
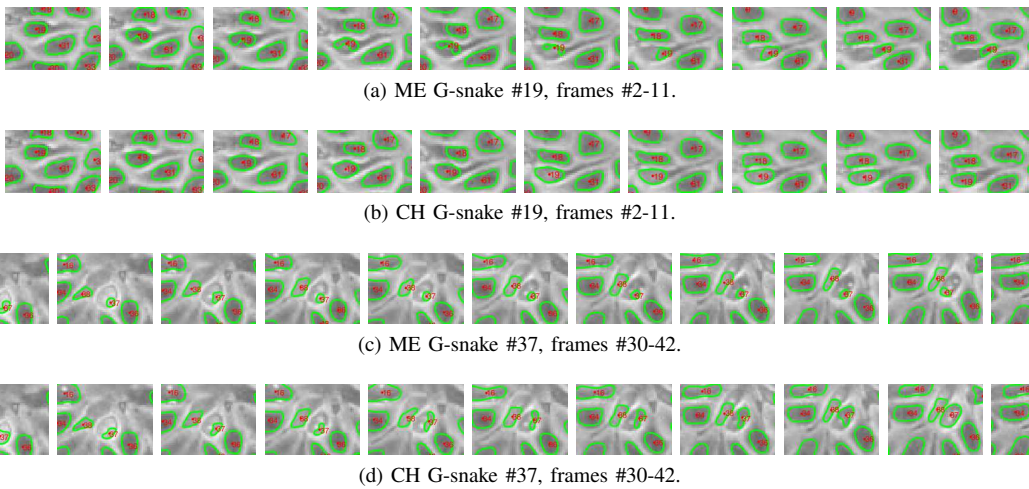

(d) CH G-snake #37, frames #30-42.

Fig. 10: Performance comparison between the ME and CH based G-snakes on the same cells. For the cells #19 and #37 (in MDCK1), the CH based G-snakes achieved better segmentation results due to the involved convexity constraint.

other parameters are maintained as the same values. Table V provides a full list of statistics that are obtained by applying different templates based G-snake models to the four image sequences. The best accuracies of the G-snake model, by

TABLE III: Quantitative results of the self-crossing events

	G-snake			Regular snake	
	ME	MB	CH	w/o re-init	w re-init
WH-PEC-d1	0	0	0	420(27)	124(27)
WH-PEC-d5	0	0	0	219(27)	83(11)
MDCK1	1(1)	7(3)	0	1185(70)	716(40)
MDCK2	1(1)	4(3)	0	1224(70)	652(37)

TABLE IV: Evaluating the ED constraint

	WH-PEC-d5	MDCK1	MDCK2
ME G-snake	93.15%	93.59%	93.18%

TABLE V: Quantitative results for tracking accuracies, with varied κ

		κ				
		0.2	0.4	0.6	0.8	1
ME	WH-PEC-d1	95.89%	**97.26%**	97.26%	95.89%	94.52%
	WH-PEC-d5	**95.89%**	95.89%	95.89%	95.89%	94.52%
	MDCK1	92.31%	94.87%	94.87%	**96.15%**	96.15%
	MDCK2	95.45%	95.45%	**96.59%**	95.45%	95.45%
MB	WH-PEC-d1	94.52%	95.89%	**97.26%**	97.26%	95.89%
	WH-PEC-d5	91.78%	94.52%	**94.52%**	94.52%	94.52%
	MDCK1	93.59%	94.87%	**97.44%**	96.15%	96.15%
	MDCK2	**95.45%**	94.32%	94.32%	95.45%	94.32%
CH	WH-PEC-d1	94.52%	**97.26%**	97.26%	97.26%	97.26%
	WH-PEC-d5	94.52%	**94.52%**	94.52%	93.15%	94.52%
	MDCK1	92.31%	93.59%	94.87%	**96.15%**	96.15%
	MDCK2	95.45%	97.73%	97.73%	96.59%	**97.73%**
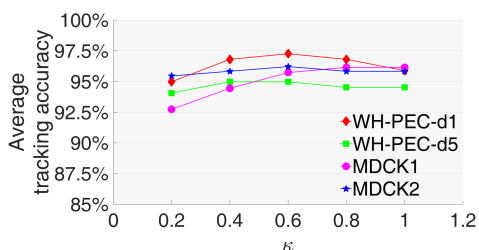
Fig. 11: The average tracking accuracies of the G-snake model based on different dynamic templates. The experiments are conducted by manually adapting $\kappa \in \{0.2, 0.4, 0.6, 0.8, 1\}$.

adapting the κ value, are highlighted in the table. Although κ is varied from 0.2 to 1, all of the tracking accuracies are successfully maintained in high and steady rates. Fig. 11 intuitively illustrates the average performance of the G-snake model based tracking. Even with a relatively small κ that gives a less competitive guiding force, the tracking accuracies are only slightly decreased.

As can be seen in Fig. 12, by lowering down the κ value, the average numbers of the SCEs are mildly increased. Peak numbers are at $\kappa = 0.2$. However, the largest SCE number is not exceeding 5% of the average SCE number with the traditional model. When κ is above 0.6, the SCEs are almost excluded.
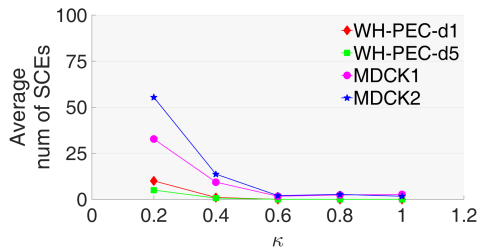
Fig. 12: The average numbers of the SCEs according to the G-snake model, with $\kappa \in \{0.2, 0.4, 0.6, 0.8, 1\}$. Each curve shows how the average number of the SCEs changes with a varied value of κ.

V. EXPERIMENTS AND COMPARISON

A. Modern Active Contours Dealing with Self-intersection

In [33], a so-called, contour-simplicity (CS) term is proposed that penalises self-intersected curves for geodesic active contour, which is a model closely related to parameteric snakes. Assuming control points on a simple closed contour are distributed in a clockwise order, inverted loops are considered as those following the anti-clockwise. To facilitate a later discussion, the explicit CS term of [33] is introduced as below,

$$E^{cs}(C) = - \sum_{(u,v) \in IS(C)} \int_u^v \frac{x(s)y'(s) - x'(s)y(s)}{2} ds \quad (10)$$

where $IS(C)$ denotes the set of ordered pairs of curve positions (u, v) s.t. $u < v$, that describe inverted segments (either closed or open). In fact, E^{cs} here is a simplified expression for the original Eqn. (11) in [33]. This is achieved by treating single or double types of inverted loops equally, since they both follow anti-clockwise orders. Note in [33] collision points are classified as positive/negative crossings, for distinguishing single/double inverted loops. Taking the derivative of Eqn. (10), the associated CS force can be written as,

$$F^{cs}(C(s)) = \begin{cases} N(s) & \text{if } s \in IS(C), \\ 0 & \text{if } s \notin IS(C). \end{cases} \quad (11)$$

which presents an interesting formula, since the potential CS force is decided by the normal vector $N(\cdot)$, and s inside (resp. not inside) the set $IS(C)$ identifies where to switch on (resp. off) the CS force (along the contour). From the implementation, this is also reasonable: evolving an inverted loop along the local normal directions will make the loop smaller or disappear, considering the inside/outside of the loop is reversed.

Also, note the algorithm of [33] is developed for the interactive segmentation tasks. Here, we directly employ the CS force and combine that with the regular snake. By that means, the constructed model (hereinafter CS-snake) is more convenient for a joint segmentation and tracking task. The following experiments compare the CS-snake and the proposed G-snake, particularly in dealing with the problem of contour self-intersection. From the energy functional aspect, the CS-snake consists of the same terms as defined in Eqn. (8), only with the GD term replaced by the CS term. And, all the
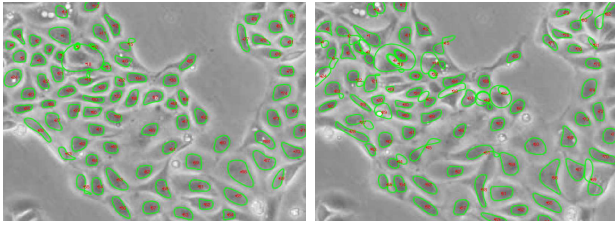
Fig. 13: Illustrating CS-snakes tested on the MDCK1 image sequence. Left to right: frames #4 and #10. Note that the CS-snake model is constructed by borrowing the contour-simplicity term from [33]).

weighting values of the CS-snake are set as the same as the G-snake, except the CS-force weight is tuned.

Fig. 13 presents the tracking results of the CS-snakes on frames #4 and #7 of the MDCK1 sequence. The events of contour self-intersection become frequent after a small number of frames. Recall that G-snakes are able to track over a larger number of frames, without developing self-loops, as previously demonstrated in Fig. 9. In addition, contour expanding is observed in CS-snakes, such as #16, #18 and #49 in frame #10 (see also Fig. 13). In order to find inducing factors for those undesirable events, additional experiments are conducted in supplementary materials, where CS-snakes are tested on synthetic images.

According to our experimental findings, the method that starts with classifying inverted loops or self-crossing knots can be highly unreliable, either for relatively simple or complex self-crossings. Also, evolving the contour by blindly following the local normal directions can be dangerous. Although our approach also divides snaxels into two groups (to determine which snaxels need the FP mechanism), there are key differences to note. First, all snaxels are treated as individuals in our classification, with no need to further group them into sub-loops or contour segments. This avoids the complexity of resolving orders among different snaxels. And, we do not evaluate clockwise/anti-clockwise loops for deciding upon the inside/outside of the snake region, since that can be confusing. Instead, by exploiting the snake-covered region, the inside/outside can be more reliably determined.

Although [17] provides an elegant way for determining the occurrence of self-crossing events, the algorithm does not deal with (automated) contour untying. In addition, the authors resort to extra strategies to locate self-crossing knots, and then they choose to split or reorder the contour. By contrast, our approach needs no explicit localization for collision points. For any given contour, our first concern is not whether a self-crossing occurs or not. Instead, we encourage the snake to obey two intuitive rules, namely the simple-curve topology and the even distribution of control points, which are unified into the proposed guiding-force mechanism. The essential idea of our method is to always provide a good template/example for the snake.

In [23], algorithmic innovations for a so-called non-intersecting force (NIF) are developed for snakes. Essentially, a volcano force model is employed that grants a stronger push force for snaxels that are close to the centerline (skeleton) of the snake. Unfortunately, the contour's self-intersection problem is not excluded by NIF snakes, regardless of using centerline or (extended) full skeleton based NIFs. Experimental tests are also included as supplementary materials. We find that self-intersection or points collision can happen at random locations along the contour, due to the fact that snaxels relatively far away from the endpoints or the core structure of the skeletons are less affected by the NIF. While, increasing the NIF weight or diffusing the NIF into a larger range give rise to the side effect of contour expansion. As long as the snake has not converged onto desirable features or positions, the volcano force (driven by the skeleton structure) might harm the contour convergence.

Most recently, Barbu *et al.* developed a method for eliminating contour self-intersection in the initialization step of a level-set active contour. The basic idea of [37] can be viewed as grouping (short) edges with orientation compatibility into a simple curve, given start and end points. However, contour grouping is by itself an unsolved problem. Also, the extension of their approach to jointly tracking multiple objects is not straightforward.

### B. Comparison with Spline-based Snakes

So far, our work has been focused on points-based parametric snakes. Spline-based snakes, belonging to another branch of parametric models, are also popular due to their advantages of fewer coefficients [38], and ease of interaction. Notably, modern spline snakes have gained success at segmenting cells with different shape modes, by either interactive or automated means. See demonstrations in works [8], [39], [40]. For that reason, this section performs contrast experiments among so-called E-snakes [39], Hermite snakes [40], as well as the proposed G-snakes.

For both of the sparse and dense cell populations, snakes are initialised as average-sized circles in the first frame, which starts the full tracking process. Note that in [39] and [40], snakes are initialized by manually outlining curves around cell boundaries. Uhlmann *et al.* also allow users to specify the number of control points (NCP) for each Hermite snake. In our experiments, fixed NCPs are exploited for the spline snakes, after a number of parameter tuning. This is for the consideration that cell shapes can be quite flexible. An optimal NCP is actually not available for all snakes/cells in the view, nor for the same snake because of cell deformation. The point sampling rate (per curve span) is also tuned, to enable suitable values for different spline snakes. Other associated parameters are set as default values by referring to [39] and [40]. See Fig. 14 for the experimental results.

The snakes behave varyingly at preserving shape/structure stability, when reacting to cell deformation or neighbor interaction. The structure of E-snakes or Hermite snakes can suddenly collapse, resulting in undesirable loops or huge (random) blobs. Checking the result of the edge-based E-snakes (namely the second image in Fig. 14a), 4 out of 13 snakes have achieved inferior segmentations in the frame #5. This includes two cases of contour self-crossing (#2 and #8 snakes) and
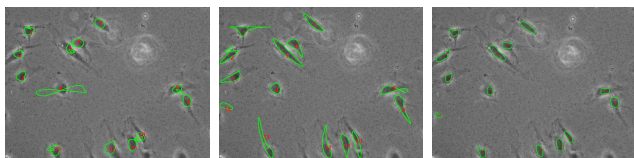
(a) From left to right: region-based E-snakes, edge-based E-snakes, and the proposed G-snakes on the same frame (#5).
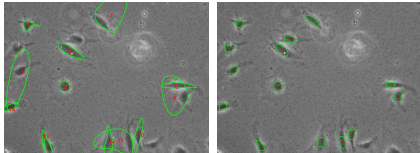
(b) Hermite snakes (left) and the proposed G-snakes (right), tracking on the same image (frame #1).

Fig. 14: Comparing the performances of E-snakes, Hermite snakes and the proposed G-snakes on tracking sparse or dense cell populations. Some snakes are not displayed, corresponding to absent (cell/snake) identity indices. This is because a notable rate of E-snakes or Hermite snakes suddenly develop huge blobs.

two cases of under-segmentation (#12 and #13 snakes). Neither region- nor edge-based E-snakes exclude contour self-intersection, as reflected in Fig. 14a. For Hermite snakes, to stably handle cells with different sizes/shapes is challenging (see Fig. 14b). The testing results of E-snakes and Hermite snakes on tracking dense populations are not presented, due to that even higher degrees of contour instabilities are observed. For the spline-based snakes, good initialization and the optimal number of control points are usually required (preferably for each snake). This makes the models less feasible, particularly in tracking cells with flexible shapes and deformations.

Still, other works may improve the robustness of active contours in object tracking. Researchers usually aim to include specific prior knowledge about classes of objects, and also they tend to use probability methods for describing families of plausible shapes. Unlike existing works, no particular shapes are enforced in our model, while strong contour stability and performance consistency have been achieved, thanks to the proposed guiding-force mechanism.

C. Comparison with Existing Cell-Tracking Works

Since existing works are commonly evaluated on different cellular datasets, and associated statistics are not always reported in the literature, qualitative and quantitative comparisons are thus jointly made in this section. See Table VI.

Comparing with [12], [15], [41], [42], the developed system has achieved higher or comparable accuracies. To facilitate fair comparison, the statistics are precision values, i.e., the number of correct (automatically) tracked cells divided by the total number of cells as automatically identified by the corresponding system or approach. Also noting that, the cell-tracking accuracies in our work are returned by uniformly fixing the guiding-force weight $\kappa = 1$. [12], [15], [41] share a similar level of cell density and spatial resolution

to us. However, we achieved higher accuracies than [12] and [41], in spite of our datasets containing more complex cell dynamics, particularly in the MDCK1 and the WH-PEC-d5 datasets. Although a comparable accuracy is reported by [15], it is worth noting postprocessing steps as employed for reconnecting broken or early-terminated trajectories. Our work requires neither postprocessing nor mid-term re-initalization for the cell trackers during the whole tracking process. Among different types of imaging techniques, fluorescence microcopy images usually have higher contrast. It is well known that phase-contrast datasets have image artifacts, and blurred cell boundaries that add considerable difficulties to image segmentation. The works [11], [27], [42], [43] employ relatively low-contrast images but take much sparser cell populations as inputs. As the density of the cell population goes down, the neighborhood-interference index and the tracking ambiguity are correspondingly decreased. So, a sparse population provides an advantage in practical tracking. Except for using sparser populations, existing works usually require the image sequence to have a relatively high frame rate, where cells move or deform slowly along the temporal axis. By contrast, our experimental datasets include relatively large-scale motions between consecutive frames (corresponding to a low frame rate).

Since dense cell-population tracking is by itself a complicated problem with many particular challenges, existing approaches usually resort to combinations of models and algorithms, and/or training processes. To temporally limit the tracking difficulty, different focuses or assumptions are made in the literature. That makes it difficult to fully implement different systems or recover equal environments for comparison. However, it is worth emphasizing that a sufficiently generalised cell-tracking system is developed, where the core algorithm is the proposed G-snake model. It is encouraging to see the proposed approach has demonstrated comparable or improved tracking accuracies, compared with the current state-of-the-art.

VI. Conclusion

In this work, a novel DSP G-snake model is proposed that solves the topology and parametrisation flaws that occur with the traditional snake model. To the best knowledge of the authors, it is the first time that a dynamic shape prior is introduced to the parametric ACM field. In addition, the following issues are efficiently tackled or solved: the contour's self-intersection problem, and the common effect that the snaxels improperly distribute along the contour. Thanks to the proposed guiding-force mechanism, the model stability and performance has been significantly enhanced. Different sets of experiments have been carried out, in order to compare the proposed model with existing active contours and related extensions. It is confirmed that the role of the proposed G-snake model cannot be replaced by either snakes that are simply re-sampled (at each iteration) or other modern competitors. Moreover, the proposed G-snake is combined with other existing forces and applied to a very challenging problem of tracking dense biological cell populations. It is confirmed that

TABLE VI: Comparison against Existing Cell-Tracking Systems

Refs	Image Mode	Image Contrast	Cell Density	Accuracy (max)	Accuracy (min)
[41]	Confocal laser scan	Low	Dense	83.9%	83.9%
[12]	Phase-contrast	Low	Dense	97.4%	81.8%
[11]	Bright-field	Medium	Sparse	n.a	n.a
[42]	Phase-contrast	Low	Sparse	87.3%	84.4%
[15]	Confocal laser scan	Low	Dense	97.0%	**96.8**%
[43]	Phase-contrast	Low	Sparse	n.a	n.a
Proposed	Phase-contrast	Low	Dense	**97.7**%	94.5%

the system developed has resulted in a significant increase in the cell tracking accuracy. Compared with existing state-of-the-art works, the proposed approach has demonstrated improved performance, in spite of application to challenging dense populations with cells that have more complex motions and larger displacements than are commonplace in the literature.

APPENDIX A
COMPARISON WITH SNAKES RE-SAMPLED AT EACH ITERATION

In theory, directly resampling the snake, according to a DSP template, is also able to remove self-intersections. So, experiments are necessary to compare the proposed G-snake model against regular snakes with resampling at each iteration (RAEI). Recall the snake contour, in the first graph of Fig. 5, is actually contracted with respect to the original, after a direct reparameterization. In order to resist the region-shrinkage effect, another set of experiments use classic snakes that are resampled according to dilated ME templates. To be specific, two steps are carried out at each iteration: First, new control points are evenly extracted along the boundary of a regular ME template or a dilated one; Then, the gradient-descent method is conducted to evolve the resampled contour. The two types of RAEI models are hereinafter referred to as ME or dilated-ME RAEI snakes. They are compared with the proposed G-snakes on tracking dense/sparse cell populations, with the experimental results shown in Fig. 15.

According to the experiments, ME RAEI snakes are gradually shrinking from frame to frame. See the segmentation failures even in the frame #3, as shown in the first column of Fig. 15. On the other hand, the dilated-ME RAEI snakes are quickly expanding into nearby regions or background (see the images in the second column). Since cell boundaries in phase-contrast images can be flexible decisions, even by human experts, one may consider some cells get tolerable segmentations by dilated-ME RAEI snakes in the dense-population case. However, a dilated-template based resampling grants expanding forces with full privilege. The contour expanding thus dominates the evolution process, and overwhelms other external forces of the snakes. This is not only severe for cells with blurred boundaries, but also for crisp boundaries. No matter whether the contour position is close enough to salient features, the expansion or shrinkage "force" (inserted at each iteration) might harm contour convergence. By contrast, the G-snake contours better conform to cell boundaries, for either
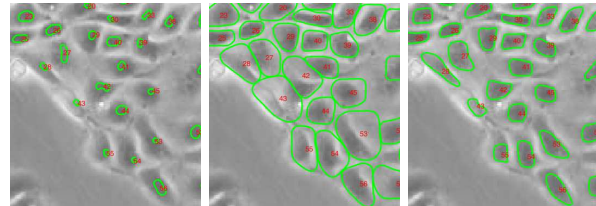
(a) ME RAEI snakes, Dilated-ME RAEI snakes, and G-snakes on dense cells tracking in frame #3.
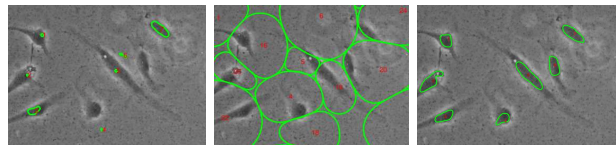
(b) ME RAEI snakes, Dilated-ME RAEI snakes, and G-snakes on tracking sparse cells in frame #12.

Fig. 15: From left to right, the segmentation and tracking results sequentially correspond to ME RAEI snakes, dilated-ME RAEI snakes and G-snakes.

dense or sparse cell populations (see the last column in the figure).

In terms of computation time, the G-snake model takes slightly increased time, in comparison to classical models with RAEI. The extra computation is mainly due to the involvement of guiding-force vectors, with a time complexity of $O(n)$ (n denoting the number of the sampled points). This process could be less trivial than existing works that enforce particular shape constraints, or it is at least similarly trivial to adding new external/internal forces.

REFERENCES

[1] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[2] F. Leymarie and M. D. Levine, "Tracking deformable objects in the plane using an active contour model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 6, pp. 617–634, 1993.

[3] C. Zimmer, E. Labruyere, V. Meas-Yedid, N. Guillen, and J.-C. Olivo-Marin, "Segmentation and tracking of migrating cells in videomicroscopy with parametric active contours: A tool for cell-based drug testing," *IEEE Trans. Med. Imaging*, vol. 21, no. 10, pp. 1212–1221, 2002.

[4] E. Jurrus, T. Tasdizen, P. Koshevoy, P. T. Fletcher, M. Hardy, C. bin Chien, W. Denk, and R. Whitaker, "Axon tracking in serial block-face scanning electron microscopy," *Medical Image Analysis*, vol. 13, no. 1, pp. 180–188, Feb. 2009.

[5] B. C. Arvind, S. K. Nagaraj, C. S. Seelamantula, and S. S. Gorthi, "Active-disc-based kalman filter technique for tracking of blood cells in microfluidic channels," in *Image Processing (ICIP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 3394–3398.

[6] J. H. Kumar, A. K. Pediredla, and C. S. Seelamantula, "Active discs for automated optic disc segmentation," in *Signal and Information Processing (GlobalSIP), 2015 IEEE Global Conference on*. IEEE, 2015, pp. 225–229.

[7] N. Ray and S. T. Acton, "Motion gradient vector flow: an external force for tracking rolling leukocytes with shape and size constrained active contours," *IEEE Trans. Med. Imaging*, vol. 23, no. 12, pp. 1466–1478, 2004.

[8] P. Thevenaz, R. Delgado-Gonzalo, and M. Unser, "The ovuscule," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 2, pp. 382–393, 2011.

[9] K. Pan, A. Kokaram, K. Gilmore, M. J. Higgins, R. Kapsa, and G. G. Wallace, "Cellsnake: A new active contour technique for cell/fibre segmentation," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*. IEEE, 2011, pp. 2153–2156.

[10] R. S. Zou and C. Tomasi, "Deformable graph model for tracking epithelial cell sheets in fluorescence microscopy," *IEEE transactions on medical imaging*, vol. 35, no. 7, pp. 1625–1635, 2016.

[11] K. E. Magnusson, J. Jaldén, P. M. Gilbert, and H. M. Blau, "Global linking of cell tracks using the viterbi algorithm," *IEEE transactions on medical imaging*, vol. 34, no. 4, pp. 911–929, 2015.

[12] K. Thirusittampalam, M. Hossain, O. Ghita, and P. Whelan, "A novel framework for cellular tracking and mitosis detection in dense phase contrast microscopy images," *Biomedical and Health Informatics, IEEE Journal of*, vol. 17, no. 3, pp. 642–653, May 2013.

[13] E. Meijering, O. Dzyubachyk, I. Smal *et al.*, "Methods for cell and particle tracking," *Methods in enzymology*, vol. 504, no. 9, pp. 183–200, 2012.

[14] R. H. Chisholm, T. Lorenzi, and J. Clairambault, "Cell population heterogeneity and evolution towards drug resistance in cancer: biological and mathematical assessment, theoretical treatment optimisation," *Biochimica et Biophysica Acta (BBA)-General Subjects*, vol. 1860, no. 11, pp. 2627–2645, 2016.

[15] M. Liu, Y. He, Y. Wei, and P. Xiang, "Plant cell tracking using kalman filter based local graph matching," *Image Vision Comput.*, vol. 60, pp. 154–161, 2017.

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

[17] A. Nakhmani and A. Tannenbaum, "Self-crossing detection and location for parametric active contours," *IEEE Transactions on Image Processing*, vol. 21, no. 7, pp. 3150–3156, 2012.

[18] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models-their training and application," *Computer vision and image understanding*, vol. 61, no. 1, pp. 38–59, 1995.

[19] T. Heimann and H.-P. Meinzer, "Statistical shape models for 3d medical image segmentation: a review," *Medical image analysis*, vol. 13, no. 4, pp. 543–563, 2009.

[20] E. Erdil, M. U. Ghani, L. Rada, A. O. Argunsah, D. Unay, T. Tasdizen, and M. Cetin, "Nonparametric joint shape and feature priors for image segmentation," *IEEE Transactions on Image Processing*, 2017.

[21] C. E. Smith and H. Schaub, "Efficient polygonal intersection determination with applications to robotics and vision," in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*. IEEE, 2005, pp. 3890–3895.

[22] L. Ji and H. Yan, "Loop-free snakes for highly irregular object shapes," *Pattern Recognition Letters*, vol. 23, no. 5, pp. 579–591, 2002.

[23] S. Kang, C.-Y. Lee, M. Gonçalves, A. D. Chisholm, and P. C. Cosman, "Tracking epithelial cell junctions in c. elegans embryogenesis with active contours guided by sift flow," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 4, pp. 1020–1033, 2015.

[24] C. Le Guyader and L. A. Vese, "Self-repelling snakes for topology-preserving segmentation models," *IEEE Transactions on Image Processing*, vol. 17, no. 5, pp. 767–779, 2008.

[25] F. Ségonne, "Active contours under topology control—genus preserving level sets," *International Journal of Computer Vision*, vol. 79, no. 2, pp. 107–117, 2008.

[26] Y. Bai, X. Han, and J. Prince, "Geometric deformable models," in *Handbook of Biomedical Imaging*. Springer, 2015, pp. 83–104.

[27] M. Möller, M. Burger, P. Dieterich, and A. Schwab, "A framework for automated cell tracking in phase contrast microscopic videos based on normal velocities," *J. Vis. Comun. Image Represent.*, vol. 25, no. 2, pp. 396–409, Feb. 2014.

[28] E. Goceri, B. Goksel, J. B. Elder, V. K. Puduvalli, J. J. Otero, and M. N. Gurcan, "Quantitative validation of anti-ptbp1 antibody for diagnostic neuropathology use: Image analysis approach," *International journal for numerical methods in biomedical engineering*, vol. 33, no. 11, 2017.

[29] J. Pedrosa, S. Queirós, O. Bernard, J. Engvall, T. Edvardsen, E. Nagel, and J. D'hooge, "Fast and fully automatic 3d left ventricular segmentation using shape-based b-spline explicit active surfaces," in *Ultrasonics Symposium (IUS), 2017 IEEE International*. IEEE, 2017, pp. 1–1.

[30] E. Göçeri, "Fully automated liver segmentation using sobolev gradient-based level set evolution," *International journal for numerical methods in biomedical engineering*, vol. 32, no. 11, 2016.

[31] S. Agarwala, D. Nandi, A. Kumar, A. K. Dhara, S. B. T. A. Sadhu, and A. K. Bhadra, "Automated segmentation of lung field in hrct images using active shape model," in *Region 10 Conference, TENCON 2017-2017 IEEE*. IEEE, 2017, pp. 2516–2520.

[32] D. Cremers, C. Schnorr, and J. Weickert, "Diffusion-snakes: combining statistical shape knowledge and image information in a variational framework," in *Variational and Level Set Methods in Computer Vision, 2001. Proceedings. IEEE Workshop on*. IEEE, 2001, pp. 137–144.

[33] J. Mille, S. Bougleux, and L. D. Cohen, "Combination of piecewise-geodesic paths for interactive segmentation," *International Journal of Computer Vision*, vol. 112, no. 1, pp. 1–22, 2015.

[34] S. Yu and Y. Lu, "Joint segmenting and tracking densely packed cells using dynamic-gvf based snakes," in *Signal Processing & Its Applications (CSPA), 2018 IEEE 14th International Colloquium on*. IEEE, 2018, pp. 113–117.

[35] J. Cheng and S. W. Foo, "Dynamic directional gradient vector flow for snakes," *IEEE Transactions on Image Processing*, vol. 15, no. 6, pp. 1563–1571, 2006.

[36] K. J. Salaycik, C. J. Fagerstrom, K. Murthy, U. S. Tulu, and P. Wadsworth, "Quantification of microtubule nucleation, growth and dynamics in wound-edge cells," *Journal of cell science*, vol. 118, no. 18, pp. 4113–4122, 2005.

[37] A. Barbu, "A directed graph approach to active contours," in *Image Processing (ICIP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 71–75.

[38] R. Delgado-Gonzalo, V. Uhlmann, D. Schmitter, and M. Unser, "Snakes on a plane: A perfect snap for bioimage analysis," *IEEE Signal Processing Magazine*, vol. 32, no. 1, pp. 41–48, 2015.

[39] R. Delgado-Gonzalo and M. Unser, "Spline-based framework for interactive segmentation in biomedical imaging," *Irbm*, vol. 34, no. 3, pp. 235–243, 2013.

[40] V. Uhlmann, J. Fageot, and M. Unser, "Hermite snakes with control of tangents," *IEEE Transactions on Image Processing*, vol. 25, no. 6, pp. 2803–2816, 2016.

[41] A. Chakraborty and A. K. Roy-Chowdhury, "Context aware spatiotemporal cell tracking in densely packed multilayer tissues," *Medical image analysis*, vol. 19, no. 1, pp. 149–163, 2015.

[42] M. A. A. Dewan, M. O. Ahmad, and M. N. S. Swamy, "Tracking biological cells in time-lapse microscopy: An adaptive technique combining motion and topological features," *IEEE Trans. Biomed. Engineering*, vol. 58, no. 6, pp. 1637–1647, 2011.

[43] T. He, H. Mao, J. Guo, and Z. Yi, "Cell tracking using deep neural networks with multi-task learning," *Image and Vision Computing*, vol. 60, pp. 142–153, 2017.
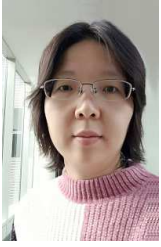
Sha Yu received her Ph.D degree in Electronic Enineering from Dublin City University, Ireland, in 2015. After that, she joined the Computational Medical Imaging Laboratory as a postdoctoral researcher, at Sun Yat-sen University, Guangzhou, China. She is currently a researcher investigator at the Insight Centre for Data Analytics, Dublin City University, Ireland. Sha's research has been supported by National Biophotonics&Imaging Platform Ireland, National Science Foundation of China, and Science Foundation Ireland. She won Best Paper Award in 14th IMVIP Conference in 2010. She graduated with First Class Honor for her master degree in Dublin City University, in 2011. She was awarded Student Travel Grant with IEEE VCIP Conference in 2013. She was awarded Best Oral Presentation in 2nd ICIAI conference in 2018. Her current research interests are in the areas of image segmentation/registration, crowded population modelling/tracking, stereo vision and 3D scene reconstruction.
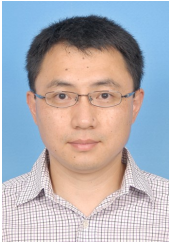
Yao Lu received the B.S. degree in mathematics from the University of Science and Technology of China, Hefei, China, in 2000. He went to graduate schools of Institute of Mathematics, Chinese Academy of Sciences and Syracuse University, Syracuse, NY, USA. He obtained his Ph.D. degree in Mathematics from Syracuse University, Syracuse, NY in 2009. He is currently a Full Professor at School of Data Science and Computer Science, Sun Yat-sen University, Guangzhou, China. Prior to his current employment, he was a research investigator at Department of Radiology, University of Michigan, Ann Arbor, MI, USA.

His research interests are in medical imaging, image processing, medical image analysis, and ill-posed problems. His research has been supported by National Science Foundation of China, Department of Education of China, and Department of Science and Technology of Guangdong Province, China. He has received awards from Recruitment Program of Global Youth Experts of Chinese Department of Organization and from Hundreds Talents Program of Sun Yet-Sen University, Guangzhou, China. He has published more than 50 journal papers and conference proceeding papers.

Derek Molloy is an Associate Professor in the School of Electronic Engineering, Faculty of Engineering & Computing at Dublin City University, Ireland, where his research and teaching focus on computer and machine vision, connected embedded systems, and 3D visualization. He is best known for his Wiley monographs on embedded Linux that are used on engineering programmes in universities across the world. He is currently working on research that involves the integration of vision systems and next generation Internet of Things (IoT) frameworks for MPSoCs.