Unsupervised Contour Tracking of Live Cells by Mechanical and Cycle Consistency Losses

Junbong Jang,
KAIST

Kwonmoo Lee[*],
Boston Children's Hospital, Harvard Medical School

Tae-Kyun Kim[*]
KAIST, Imperial College London

Abstract

Analyzing the dynamic changes of cellular morphology is important for understanding the various functions and characteristics of live cells, including stem cells and metastatic cancer cells. To this end, we need to track all points on the highly deformable cellular contour in every frame of live cell video. Local shapes and textures on the contour are not evident, and their motions are complex, often with expansion and contraction of local contour features. The prior arts for optical flow or deep point set tracking are unsuited due to the fluidity of cells, and previous deep contour tracking does not consider point correspondence. We propose the first deep learning-based tracking of cellular (or more generally viscoelastic materials) contours with point correspondence by fusing dense representation between two contours with cross attention. Since it is impractical to manually label dense tracking points on the contour, unsupervised learning comprised of the mechanical and cyclical consistency losses is proposed to train our contour tracker. The mechanical loss forcing the points to move perpendicular to the contour effectively helps out. For quantitative evaluation, we labeled sparse tracking points along the contour of live cells from two live cell datasets taken with phase contrast and confocal fluorescence microscopes. Our contour tracker quantitatively outperforms compared methods and produces qualitatively more favorable results. Our code and data are publicly available at https://github.com/JunbongJang/contour-tracking/

1. Introduction

During cell migration, cells change their morphology by expanding or contracting their plasma membranes continuously like viscoelastic materials [21]. The dynamic change in the morphology of a live cell is called cellular morphodynamics and ranges from cellular to the subcellular movement of contour at varying spatiotemporal scales. While cellular morphodynamics plays a vital role in angiogenesis, immune response, stem cell differentiation, and cancer invasiveness [6, 17], it is challenging to understand the various

---

[*]Corresponding authors: kimtaekyun@kaist.ac.kr and kwonmoo.lee@childrens.harvard.edu.

functions of cellular morphodynamics because its uncharacterized heterogeneity could mask crucial mechanistic details. As an initial step to understanding cellular morphodynamics, cellular morphodynamics is quantified by tracking every point along the cellular contour (contour tracking) and estimating their velocity [12, 13, 21]. Then, quantification of cellular morphodynamics is further processed by other downstream machine learning tasks to characterize the drug-sensitive morphodynamic phenotypes with distinct molecular mechanisms [6,20,36]. Because contour tracking (e.g., Fig. 1) is the important first step, the tracking accuracy is crucial in this live cell analysis.

There are two main difficulties involved with contour tracking of a live cell. First, the live cell's contour exhibits visual features that can be difficult to distinguish by human eyes, meaning that a pixel and its neighboring pixels have similar color values or features. Optical flow [14, 31] can track every pixel in the current frame by assuming that the corresponding pixel in the next frame will have the same distinct feature, but this assumption is not sufficient to find corresponding pixels given cellular visual features. Second, the expansion and contraction of the cellular contour change the total number of tracking points due to one point splitting into many points or many points converging into one. PoST [24] tracks a fixed number of a sparse set of points that cannot accurately represent the fluctuating shape of the cellular contour. Other deep contour tracking or video segmentation methods [10, 27, 39] do not provide dense point-to-point correspondence information between a contour and its next contour.

Previous cellular contour tracking method (mechanical model) [21] evades the first problem by taking the segmentation of the cell body as inputs instead of raw images. Then, it finds the dense correspondences of all points between two contours by minimizing the normal torsion force and linear spring force with the Marquard-Levenberg algorithm [23]. However, the mechanical model has limited accuracy because it does not consider visual features in raw images. Also, its linear spring force which keeps every distance between points the same is less effective during the expansion and contraction of the cell, as shown in our experiments (see Tab. 1).

Therefore, we present a deep learning-based contour tracker that can overcome these difficulties. Our contour tracker is comprised of a feature encoder, two cross attentions [35], and a fully connected neural network (FCNN) for offset regression, as shown in Fig. 2. Given two consecutive images and their contours represented as a sequence of points, our contour tracker encodes the visual features of two images and samples their feature at the location of contours. The sampling makes our contour tracker focus on contour features and reduces the noise from irrelevant features unlike optical flow [14]. The cross attention [35] fuses the sampled features from two contours globally and locally and regresses the offset for each contour point of the first frame. To obtain the dense point-to-point correspondences between the current and the next contours, offset points from the current contour are matched with the closest contour points in the next frame. In every frame, some contour points merge due to contraction, so new contour points emerge in the next frame as shown in Fig. 1. With dense point-to-point correspondences, new contour points in the next contour are also tracked. The proposed architectural design achieves the best accuracy

among variants, including circular convolutions [26], and correspondence matrix [4]. To the best of our knowledge, this is the first deep learning-based contour tracking with dense point-to-point correspondences for live cells.

In this contour tracking, supervised learning is not feasible because it is difficult to label every point of the contour manually. Instead, we propose to train our contour tracker solely by unsupervised learning comprised of mechanical and cycle consistency losses. Inspired by the mechanical model [21] that minimizes the normal torsion and linear spring force, we introduce the mechanical losses to end-to-end learning. The mechanical-normal loss that keeps the angle difference small between the offset point and the direction normal to the cellular contour played a significant role in boosting accuracy. Also, we implement cycle consistency loss to encourage all contour points tracked forward-then-backward to return to their original location. However, previous approaches such as PoST [24] and Animation Transformer (AnT) [4] rely on supervised learning in addition to cycle consistency loss or find mid-level correspondences [38] instead of pixel-level correspondences.

We evaluate our contour tracker on the live cell dataset taken with a phase contrast microscope [13] and another live cell dataset taken with a confocal fluorescence microscope [36]. For a quantitative comparison of contour tracking methods, we labeled sparse tracking points on the contour of live cells for all sampled frames. In total, we labeled 13 live cell videos for evaluation. Evaluation with a sparse set of points is motivated by the fact that if tracking of dense contour points is accurate, tracking any one of contour points should be accurate also. We also qualitatively show our contour tracker works on another viscoelastic organism, jellyfish [30]. Our contributions are summarized as follows.

- We propose the first deep learning-based model that tracks cellular contours densely while surpassing the accuracy of other methods.

- We present an unsupervised learning strategy by mechanical loss and cycle consistency loss for contour tracking.

- We demonstrate that the use of forward and backward cross attention with cycle consistency has a synergistic effect on finding accurate dense correspondences.

- We label tracking points in the live cell videos and quantitatively evaluate cellular contour tracking for the first time.

2. Related Work

2.1. Tracking

Cell tracking is a method that tracks the movement of a cell as one object [32]. It first segments cells and then finds their trajectories and velocities by solving the graph of potential cell tracks with integer linear programming [28]. Another work uses coupled minimum-cost flow [25] to account for splitting and merging events of the cell. The entire population of cells can be densely tracked by the optical flow [40]. In contrast, our contour tracker deals with points along the cellular contour, which is a portion of the entire cell.

The optical flow can be used for contour tracking since it can track the movement of every pixel in the video [1]. When there is no ground truth optical flow, an unsupervised optical flow model such as UFlow [14] is used. UFlow is based on the PWC-Net [31] and trains by minimizing the photometric loss between the warped second image and the first image in pixel intensity. However, optical flow can be confused by tracking features that are not related to contour. Occasionally, visible membrane features go inside cellular interiors, hindering accurate contour tracking.

Another way to perform contour tracking is by iteratively running the contour prediction method in every frame of the video. The contour prediction method represents the object boundary by a sequence of sparse points connected with straight lines and regresses offsets for the initial set of points to fit the object boundary. There are conventional contour prediction methods such as Snake [15] or deep learning-based models such as DeepSnake [26] and DANCE [19], which performs real-time objection detection. PoST [24] extends the contour prediction method to regress offsets for 128 points along the contour of an object in the current frame to get a new contour in the next frame. It is trained by supervised learning on the synthetic dataset having distinct visual features that are easily trackable by human eyes. Unlike PoST [24], our contour tracker tracks a varying number of points along the entire contour with challenging visual features.

2.2. Mechanical Model

The mechanical model [21] optimizes the nonlinear equation comprised of the normal torsion force which encourages the points to move perpendicular to the contour and the linear spring force which keeps the distance between neighboring points the same. Features in the direction normal to the contour are widely used by the active shape model [7], HMM contour tracker [5] or particle filter-based contour tracker [3] to find the matching point in the next contour. The linear spring force is similar to the first-order term in the Snake algorithm [15] which minimizes the distance between points. Also, ant colony optimization [33] improved the contour correspondence accuracy by incorporating the proximity information of neighboring points.

2.3. Dense Correspondences

Dense point correspondences between the current contour and the next contour are necessary to track each contour point throughout the video. Deformable surface tracking [11,37] finds the correspondences between a fixed number of key points on a fixed 3D surface area throughout the video by a mesh-based deformation model. Animation Transformer [4] uses cross attention [35] to find dense correspondences between two line segments. The dense correspondences are predicted as a 2D correspondence matrix, similar to the feature matching step in the 3D point cloud registration [22, 34]. ContourFlow [9] finds the point correspondences among the fragmented contours. Instead of predicting correspondences between two contours by feature matching, our contour tracker predicts the offset from current contour points to utilize mechanical loss [21]. If our contour tracker predicts the correspondence instead of offset, the computation of mechanical loss becomes non-differentiable for end-to-end learning.

3. Method

In this section, we explain our architecture and unsupervised learning strategy with mechanical and cycle consistency losses as shown in Fig. 2.

3.1. Architecture

As input, our contour tracker takes the current frame I_t and contour C_t and the next frame I_{t+1} and contour C_{t+1}. The contour is comprised of a sequence of contour points $p_t^i \in C_t$ in 2D coordinates. Every pixel along the contour extracted from the segmentation mask becomes a contour point p_t^i. The current frame I_t and the next frame I_{t+1} are encoded by ImageNet [8] pre-trained VGG16 [29] and upsampled by Feature Pyramid Network (FPN) [18] to match the size of input images, I_t and I_{t+1}. Then, image features at the location of contour points are sampled from FPN feature map. The first image's features are sampled at the location of first contour points C_t and the second image's features are sampled at the location of second contour points C_{t+1}. The sinusoidal positional embedding [35] and contour points' coordinates are concatenated to image features.

The multi-head cross attention (MHA) [35] is used to fuse the feature representation of two contours with arbitrary contour lengths and to capture global and local contour features. Our contour tracker has forward and backward cross attentions. The forward cross attention takes the first contour's feature as a query and the second contour's feature as key and value. The backward cross attention takes the second contour's feature as a query and the first contour's feature as a key and value. Lastly, FCNN comprised of 3 linear layers with ReLU activation in between receives the fused features from the forward cross attention and regresses the offset $O_{t \to t+1}$ of all contour points C_t in the first frame. The same FCNN receives the fused features from the backward cross attention and regresses the backward offset $O_{t+1 \to t}$ of all contour points C_{t+1} in the second frame. The backward offset is necessary to compute cycle consistency loss.

3.2. Unsupervised Learning

Cycle Consistency Loss.—Computing cycle consistency loss is a three-step process given two consecutive images with contour points. First, contour points in the first frame C_t move by regressing offset $O_{\to t+1}$ from their current positions. Second, each offset point is matched with the closest point on the second frame's contour C_{t+1}. This operation is denoted by ϕ. Lastly, offset points matched to the second frame's contour are tracked back to the first frame by regressing backward offsets $O_{t+1 \to t}$. Without the second step, learning by cycle consistency loss fails because the model can regress zero forward and backward offsets to obtain zero cycle consistency loss. ϕ is non-differentiable, but gradient flows to both forward and backward cross attentions because cycle consistency loss is comprised of forward and backward consistency defined as follows:

$$L_{forward} = \sum_{i=0}^{N_t - 1} \| p_t^i - O_{t+1 \to t}(\phi(O_{t \to t+1}(p_t^i))) \|_2$$

(1)

$$L_{backward} = \sum_{i=0}^{N_t - 1} \| p_{t+1}^i - O_{t \to t+1}(\phi(O_{t+1 \to t}(p_{t+1}^i))) \|_2$$

(2)

$$L_{cycle} = L_{forward} + L_{backward}$$

(3)

where N_t denotes the total number of contour points at time t because the contour can expand or contract.

Mechanical-Normal Loss.—We reformulate the normal force in the mechanical model [21] as follows. For each contour point, normal vectors n_t^i orthogonal to the contour C_t are numerically computed by approximating tangent vectors at each contour point by central difference and rotating tangent vectors by 90 degrees. Then, normal vectors and offsets of all contour points are normalized to unit vectors. Lastly, we compute the L1 difference between them as follows:

$$L_{\text{mech-normal}} = \sum_{i=1}^{N_t - 2} \left\| \frac{n_t^i}{\|n_t^i\|_2} - \frac{O_{t \to t+1}(p_t^i)}{\|O_{t \to t+1}(p_t^i)\|_2} \right\|_1$$

(4)

From N_t contour points, the first and last point is excluded from computation since their tangent vectors cannot be approximated.

Please refer to the supplementary section for other unsupervised learning losses. For optimal performance (see the ablation study in Tab. 1), the total training loss is a sum of the cycle consistency and mechanical-normal loss:

$$L_{total} = L_{cycle} + L_{\text{mech-normal}}$$

(5)

3.3. Differentiable Sampling

To update our network during backpropagation through the sampling, we use the bilinear sampling from UFlow [14] to retrieve the pixel intensity or image feature I_{xy} at a coordinate (x, y) as shown in Fig. 3. I_{xy} has nonzero gradients with respect to coordinates or features at four adjacent points. This method samples features located at the contour points C_t/C_{t+1} or offset points in the training and inference.

3.4. Pre-processing and Labeling

From the binary segmentation mask, the contour with one-pixel width is extracted and the contour is converted to an ordered sequence of contour points by an off-the-shelf algorithm of contour finder [2], as shown in Fig. 4(a). For instance, if the live cell is anchored to the left image border, the points are ordered starting from the leftmost top point in every frame. Points touching the image border are not considered for contour tracking. These ordered sequences of contour points do not have point-to-point correspondences.

For quantitative evaluation, we labeled five tracking points roughly equal distances apart from each other in low temporal resolution (every fifth frame of the video). However, we examined the video in 5x higher temporal resolution (every consecutive frame). For instance, to label the tracking point in the 10th frame from the 5th frame, we examined consecutive frames between the 5th and the 10th frames, as shown in Fig. 4(b). Labeling the tracking point can be ambiguous without examing those consecutive frames, given the large cellular motion. The tracking points that move outside the image boundary or become occluded due to cellular movement were not labeled. Labeled tracking points were used for evaluation only.

4. Experiments

4.1. Dataset

The confocal fluorescence dataset [36] contains 40 live cell videos taken with confocal fluorescence microscopy. They are classified into 9 different categories based on their cellular morphodynamics. Therefore, we randomly picked one video from each category to validate our contour tracker on all types of cellular morphodynamics. We trained on 31 live cell videos and validated our contour tracker on the other 9 live cell videos. Phase Contrast dataset [13] contains 4 multi-cellular live cell videos and 5 single-cellular live cell videos taken with a phase contrast microscope. We train on 5 single-cellular live cell videos and validate our contour tracker on 4 multi-cellular live cell videos.

Each live cell video is 200 frames long, and every frame is segmented in both datasets. We sampled every fifth frame from the video for contour tracking. By sampling, we use fewer frames for contour tracking and evaluate the robustness of our contour tracker in a low temporal resolution setting. For labeling tracking points, we examined all 200 frames to see each tracked point's cellular topology, visual features, and trajectory from previous consecutive frames.

4.2. Implementation Details

For training, live cell videos from the phase contrast dataset [13] are resized to 256^2, and the live cell videos from the confocal fluorescence dataset [36] are resized to 512^2. We trained our contour tracker using Adam [16] optimizer with an initial learning rate of 0.0001 and linear learning rate decay after 10k iterations. Also, we trained our contour tracker for 50k iterations with a batch size of 8 on one TITAN RTX GPU for 1 day. For inference, only the forward cross attention is used to regress offsets. The offset points are moved to the closest

contour point by operation ϕ at each frame to obtain the dense correspondences between the current contour C_t and the next contour C_{t+1}.

4.3. Evaluation Metrics

To evaluate the point tracking, we use the spatial accuracy (SA) introduced in [24] and introduce contour accuracy (CA). SA measures the distance between the ground truth points g_t and the predicted points $p_t = \phi(O_{t-1 \to t}(p_{t-1}))$. If the distance is less than a threshold τ, 1 is added. Otherwise, 0 is added. Each tracking point's x and y coordinates are normalized by image height and width such that the x and y coordinates range from 0 to 1. The contour points in the first frame $p_0 = C_0$ are tracked and evaluated against the ground truth points g_t at each time step t:

$$SA_\tau(p_t, g_t) = \lambda \sum_{t=1}^{T-1} \sum_{i=0}^{N-1} (\|p_t^i - g_t^i\|_2 < \tau)$$

(6)

where T is the total number of frames in the video, $N = 5$ is the total number of labeled tracking points and $\lambda = \frac{1}{(T-1)N}$. CA measures the arc length between two points on the contour. It is equivalent to measuring the difference between the ground truth point's indices and the predicted point's indices for contour points. Due to the fluctuating shape of the cellular contour, two points close to each other in the image space can be far apart in terms of the arc length. The arc length and spatial distance between two points are equal when the contour is a straight line. Let $C_t()$ be a function that returns the index of the contour point given the coordinate of the contour point.

$$CA_\tau(p_t, g_t) = \lambda \sum_{t=1}^{T-1} \sum_{i=0}^{N-1} (\|C_t(p_t^i) - C_t(g_t^i)\|_1 < \tau)$$

(7)

Then, CA is normalized by the total number of contour points in the current frame.

4.4. Ablation Study

The spatial accuracy (SA) and contour accuracy (CA) are measured with multiple thresholds since some models can perform better in lower thresholds while others perform better in higher thresholds.

Loss Functions.—We compare one supervised learning loss and combinations of four unsupervised learning losses in Tab. 1. For supervised learning on dense tracking points, contour point-to-point correspondences predicted by the mechanical model [21] are used as ground truth correspondences. Then, our contour tracker is trained to minimize the L2 distance between predicted points' and ground truth points' locations, similar to PoST [24].

The same architecture shown in Fig. 2 is used for ablation. Supervised learning yields lower accuracy than training with mechanical-normal loss or cycle consistency loss alone. The low accuracy can be due to inaccurate pseudo-labels or overfitting on training labels which does not generalize to new contour points in validation videos.

Unsupervised learning by the mechanical-normal loss or cycle consistency loss alone have significantly higher performance than other losses, such as photometric loss. However, adding the mechanical-linear loss to the mechanical-normal loss degrades the performance. Training with the mechanical-linear loss alone or other combinations of losses not shown in the table also yields low accuracy. Adding the mechanical-normal loss and cycle consistency loss yields the highest spatial and contour accuracy.

Architecture.—We replaced or removed a component of our architecture to see how each component contributes to the overall performance, as shown in Tab. 2. No cross attention (No Cross) fuses point features sampled from the first and the second images by adding them. Single cross attention (Single Cross) only uses one cross attention to fuse point features. The contour tracker with one cross attention outperforms the contour tracker without any cross attention in a low threshold setting but not in higher threshold settings. Single cross attention is not effective, possibly because the movement of the live cell played backward is physically different than the natural live cell movement. So contour features need to be handled differently depending on forward or backward directions. Our contour tracker using forward and backward cross attentions yields much higher accuracy than using one or zero cross attention.

From DeepSnake [26] and PoST [24], circular convolution is known to be effective for point regression given a sequence of contour points. When FCNN is replaced with circular convolution (Circ Conv), the model achieves much lower accuracy at low thresholds. Since our contour tracker handles a cellular contour with disconnected endpoints, we also test 1D convolution (1D Conv). Using circular convolution or 1D convolution decreases the accuracy at low thresholds. We chose the model with the highest accuracy at low thresholds because the model's accuracy at low thresholds reveals its pixel-level accuracy, and high pixel-level accuracy is known to yield less noisy and stronger morphodynamic patterns [13].

4.5. Comparison with Other Methods

We compare our contour tracker against mechanical model [21] and other deep learning-based methods: UFlow [14] and PoST [24]. Since pre-trained PoST without any modifications yields very low spatial and contour accuracy, we modified it to utilize features along the current C_t and the next contours C_{t+1}. Also, we trained PoST on the live cell dataset with our cycle consistency loss. For inference, both UFlow and PoST offset points $O_{t \to t+1}(p_i)$ move to the closest contour points in C_{t+1}. This is the same inference heuristic used for our contour tracker.

Phase Contrast Dataset.—Our contour tracker outperforms all the other methods in all threshold settings in Tab. 3. UFlow [14] has trouble tracking contour points with the lowest accuracy. Unlike other methods, UFlow does not use contour features and does not focus

on tracking the contour only. PoST [24] performs better than UFlow but still worse than the mechanical model or our contour tracker. Lastly, the mechanical model performs better than other deep learning models except ours.

Confocal Fluorescence Dataset.—The overall performance in Tab. 4 is lower than Tab. 3 because the confocal fluorescence dataset [36] is taken in higher resolution with fine details and contains some segmentation error from thresholding. Despite the difficulty, our contour tracker still outperforms all the other methods in all threshold settings as shown in Tab. 4. Consistent with the quantitative results, we qualitatively show in Fig. 5 that our contour tracker can track contour points with closer proximity to the ground truth labels compared to the mechanical model in both phase contrast and confocal fluorescence datasets. Please refer to the supplementary section for the visualization of a long sequence of videos.

4.6. Quantification of Morphodynamics

We quantify cellular morphodynamics of one of the phase contrast live cell videos [13] tracked by our contour tracker as a heatmap in Fig. 6. We chose two far-apart contour points such that the velocities of all contour points between those two points are measured. Only the velocity along the normal vector of contour points is considered. The red regions indicate outward motion (protrusion) from the cell body and the blue regions indicate inward motion to the cell nucleus of the cellular contour.

4.7. Contour Tracking of a Jellyfish

Our live cell videos contain live cells anchored to one of the sides of the image. In this section, we show that our contour tracker can also work on a different viscoelastic organism floating in space. We observed that jellyfish has a similar viscoelastic property to live cell, so we tested our contour tracker on Rainbow Jellyfish Benchmark from StyleGan-V [30]. We cropped the center of a video to get 512^2 patches containing a jellyfish and segmented it by thresholding. Our contour tracker can track the contour of a jellyfish with dense point-to-point correspondences as shown in Fig. 7. Please refer to the supplementary section for more details.

5. Conclusion

We present a novel deep learning-based contour tracking with correspondence for live cells and train it without any ground truth tracking points. We systematically tested various unsupervised learning strategies on top of the proposed architecture with cross attention and found that a combination of mechanical and cycle consistency losses is the best. Our contour tracker outperforms the classical mechanical model and other deep learning-based methods on phase contrast and confocal fluorescence live cell datasets. In the field of computer vision, we hope this work sheds light on a new type of object tracking (e.g. viscoelastic materials), which prior arts cannot adequately capture.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgements.

References

[1]. Balasundaram A, Ashok Kumar S, and Magesh Kumar S. Optical flow based object movement tracking. International Journal of Engineering and Advanced Technology, 9(1):3913–3916, 2019.

[2]. Bradski G. The OpenCV Library. Dr. Dobb's Journal of Software Tools, 2000.

[3]. Cao Songxiao and Wang Xuanyin. Visual contour tracking based on inner-contour model particle filter under complex background. EURASIP Journal on Image and Video Processing, 2019(1):1–13, 2019.

[4]. Casey Evan, Pérez Víctor, and Li Zhuoru. The animation transformer: Visual correspondence via segment matching. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 11323–11332, 2021.

[5]. Chen Yunqiang, Rui Yong, and Huang Thomas S. Multicue hmm-ukf for real-time contour tracking. IEEE transactions on pattern analysis and machine intelligence, 28(9):1525–1529, 2006. [PubMed: 16929738]

[6]. Choi Hee June, Wang Chuangqi, Pan Xiang, Jang Junbong, Cao Mengzhi, Brazzo Joseph A, Bae Yongho, and Lee Kwonmoo. Emerging machine learning approaches to phenotyping cellular motility and morphodynamics. Physical Biology, 18(4):041001, 2021.

[7]. Cootes Tim, Baldock ER, and Graham J. An introduction to active shape models. Image processing and analysis, 328:223–248, 2000.

[8]. Deng Jia, Dong Wei, Socher Richard, Li Li-Jia, Li Kai, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pages 248–255. Ieee, 2009.

[9]. Di Huijun, Shi Qingxuan, Lv Feng, Qin Ming, and Lu Yao. Contour flow: Middle-level motion estimation by combining motion segmentation and contour alignment. In Proceedings of the IEEE International Conference on Computer Vision, pages 4355–4363, 2015.

[10]. Elmahdy Mohamed S, Jagt Thyrza, Zinkstok Roel Th, Qiao Yuchuan, Shahzad Rahil, Sokooti Hessam, Yousefi Sahar, Incrocci Luca, Marijnen CAM, Hoogeman Mischa, et al. Robust contour propagation using deep learning and image registration for online adaptive proton therapy of prostate cancer. Medical physics, 46(8):3329–3343, 2019. [PubMed: 31111962]

[11]. Hilsmann Anna and Eisert Peter. Tracking deformable surfaces with optical flow in the presence of self occlusion in monocular image sequences. In 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pages 1–6. IEEE, 2008.

[12]. Jang Junbong, Hallinan Caleb, and Lee Kwonmoo. Protocol for live cell image segmentation to profile cellular morphodynamics using mars-net. STAR protocols, 3(3):101469, 2022. [PubMed: 35733606]

[13]. Jang Junbong, Wang Chuangqi, Zhang Xitong, Choi Hee June, Pan Xiang, Lin Bolun, Yu Yudong, Whittle Carly, Ryan Madison, Chen Yenyu, and Lee Kwonmoo. A deep learning-based segmentation pipeline for profiling cellular morphodynamics using multiple types of live cell microscopy. Cell Reports Methods, Oct 2021.

[14]. Jonschkowski Rico, Stone Austin, Barron Jonathan T, Gordon Ariel, Konolige Kurt, and Angelova Anelia. What matters in unsupervised optical flow. In European Conference on Computer Vision, pages 557–572. Springer, 2020.

[15]. Kass Michael, Witkin Andrew, and Terzopoulos Demetri. Snakes: Active contour models. International journal of computer vision, 1(4):321–331, 1988.

[16]. Kingma Diederik P. and Ba Jimmy. Adam: A method for stochastic optimization. In Bengio Yoshua and LeCun Yann, editors, 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.

[17]. Lee Kwonmoo, Elliott Hunter L., Oak Youbean, Zee Chih-Te, Groisman Alex, Tytell Jessica D., and Danuser Gaudenz. Functional hierarchy of redundant actin assembly factors revealed by fine-grained registration of intrinsic image fluctuations. Cell Systems, 1(1):37–50, 2015. [PubMed: 26273703]

[18]. Lin Tsung-Yi, Dollár Piotr, Girshick Ross, He Kaiming, Hariharan Bharath, and Belongie Serge. Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2117–2125, 2017.

[19]. Liu Zichen, Liew Jun Hao, Chen Xiangyu, and Feng Jiashi. Dance: A deep attentive contour model for efficient instance segmentation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pages 345–354, 2021.

[20]. Ma Xiao, Dagliyan Onur, Hahn Klaus M, and Danuser Gaudenz. Profiling cellular morphodynamics by spatiotemporal spectrum decomposition. PLoS computational biology, 14(8):e1006321, 2018. [PubMed: 30071020]

[21]. Machacek Matthias and Danuser Gaudenz. Morphodynamic profiling of protrusion phenotypes. Biophysical journal, 90(4):1439–1452, 2006. [PubMed: 16326902]

[22]. Min Taewon, Song Chonghyuk, Kim Eunseok, and Shim Inwook. Distinctiveness oriented positional equilibrium for point cloud registration. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 5490–5498, 2021.

[23]. Moré Jorge J. The levenberg-marquardt algorithm: implementation and theory. In Numerical analysis, pages 105–116. Springer, 1978.

[24]. Nam Gunhee, Heo Miran, Oh Seoung Wug, Lee Joon-Young, and Kim Seon Joo. Polygonal point set tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5569–5578, 2021.

[25]. Padfield Dirk, Rittscher Jens, and Roysam Badrinath. Coupled minimum-cost flow cell tracking for high-throughput quantitative analysis. Medical image analysis, 15(4):650–668, 2011. [PubMed: 20864383]

[26]. Peng Sida, Jiang Wen, Pi Huaijin, Li Xiuli, Bao Hujun, and Zhou Xiaowei. Deep snake for real-time instance segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8533–8542, 2020.

[27]. Saboo Shivam, Lefebvre Frederic, and Demoulin Vincent. Deep learning and interactivity for video rotoscoping. In 2020 IEEE International Conference on Image Processing (ICIP), pages 643–647. IEEE, 2020.

[28]. Scherr Tim, Löffler Katharina, Böhland Moritz, and Mikut Ralf. Cell segmentation and tracking using cnn-based distance predictions and a graph-based matching strategy. Plos One, 15(12):e0243219, 2020. [PubMed: 33290432]

[29]. Simonyan Karen and Zisserman Andrew. Very deep convolutional networks for large-scale image recognition. In Bengio Yoshua and LeCun Yann, editors, 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.

[30]. Skorokhodov Ivan, Tulyakov Sergey, and Elhoseiny Mohamed. Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3626–3636, 2022.

[31]. Sun Deqing, Yang Xiaodong, Liu Ming-Yu, and Kautz Jan. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 8934–8943, 2018.

[32]. Ulman Vladimír, Maška Martin, Magnusson Klas EG, Ronneberger Olaf, Haubold Carsten, Harder Nathalie, Matula Pavel, Matula Petr, Svoboda David, Radojevic Miroslav, et al. An

objective comparison of cell-tracking algorithms. Nature methods, 14(12):1141–1152, 2017. [PubMed: 29083403]

[33]. Van Kaick Oliver, Hamarneh Ghassan, Zhang Hao, and Wighton Paul. Contour correspondence via ant colony optimization. In 15th Pacific Conference on Computer Graphics and Applications (PG'07), pages 271–280. IEEE, 2007.

[34]. Van Kaick Oliver, Zhang Hao, Hamarneh Ghassan, and Cohen-Or Daniel. A survey on shape correspondence. In Computer graphics forum, volume 30, pages 1681–1707. Wiley Online Library, 2011.

[35]. Vaswani Ashish, Shazeer Noam, Parmar Niki, Uszkoreit Jakob, Jones Llion, Gomez Aidan N, Kaiser Łukasz, and Polosukhin Illia. Attention is all you need. Advances in neural information processing systems, 30, 2017.

[36]. Wang Chuangqi, Choi Hee June, Kim Sung-Jin, Desai Aesha, Lee Namgyu, Kim Dohoon, Bae Yongho, and Lee Kwonmoo. Deconvolution of subcellular protrusion heterogeneity and the underlying actin regulator dynamics from live cell imaging. Nature Communications, 9(1):1688, Apr 2018.

[37]. Wang Tao, Ling Haibin, Lang Congyan, Feng Songhe, and Hou Xiaohui. Deformable surface tracking by graph matching. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 901–910, 2019.

[38]. Wang Xiaolong, Jabri Allan, and Efros Alexei A. Learning correspondence from the cycle-consistency of time. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2566–2576, 2019.

[39]. Yeap PL, Noble DJ, Harrison Karl, Bates AM, Burnet NG, Jena Rajesh, Romanchikova M, Sutcliffe MPF, Thomas SJ, Barnett GC, et al. Automatic contour propagation using deformable image registration to determine delivered dose to spinal cord in head-and-neck cancer radiotherapy. Physics in Medicine & Biology, 62(15):6062, 2017. [PubMed: 28573978]

[40]. Zhou Felix Y, Ruiz-Puig Carlos, Owen Richard P, White Michael J, Rittscher Jens, and Lu Xin. Motion sensing superpixels (moses) is a systematic computational framework to quantify and discover cellular motion phenotypes. elife, 8:e40162, 2019. [PubMed: 30803483]
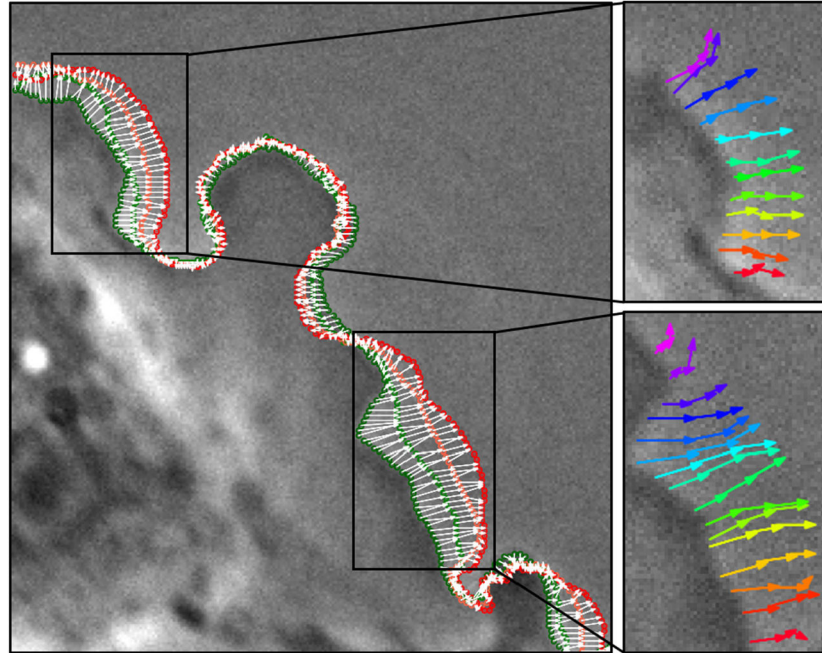
Figure 1. Visualization of contour tracking results.
Dense point correspondences between adjacent contours are shown with white arrows overlaid on the first frame. The first frame's contour points are in dark green, and the last frame's contour points are in red. Only half of the contour points and correspondences are shown for visualization purposes. The trajectories of a few tracked points are shown on the right.
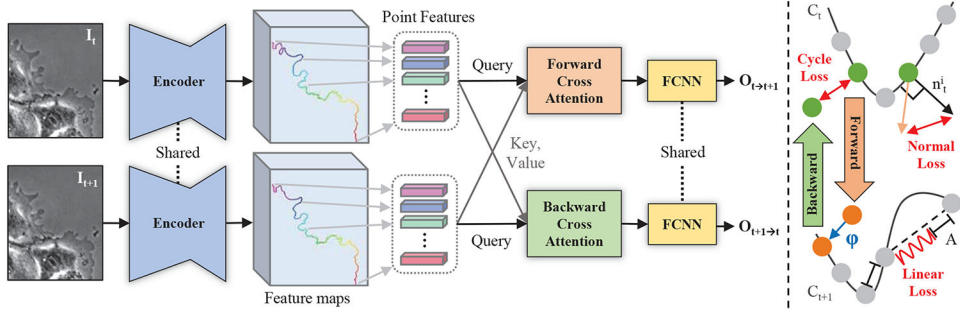
Figure 2. Our architecture on the left and unsupervised learning losses on the right.
Shared encoder comprised of VGG16 and FPN encodes first and second images. Point features are sampled at the location of ordered contour points indicated by rainbow colors from red to purple. Point features are inputted as query or key and value to the cross attentions. Lastly, shared FCNN takes the fused features and regresses forward $O_{t \rightarrow t+1}$ or backward $O_{t+1 \rightarrow t}$ offsets. The cycle consistency, mechanical-normal, and mechanical-linear losses are shown in red color.

$$w_a = (x_1-x)(y_1-y)$$
$$w_b = (x_1-x)(y-y_0)$$
$$w_c = (x-x_0)(y_1-y)$$
$$w_d = (x-x_0)(y-y_0)$$

$$I_{xy} = w_a \times I_a + w_b \times I_b$$
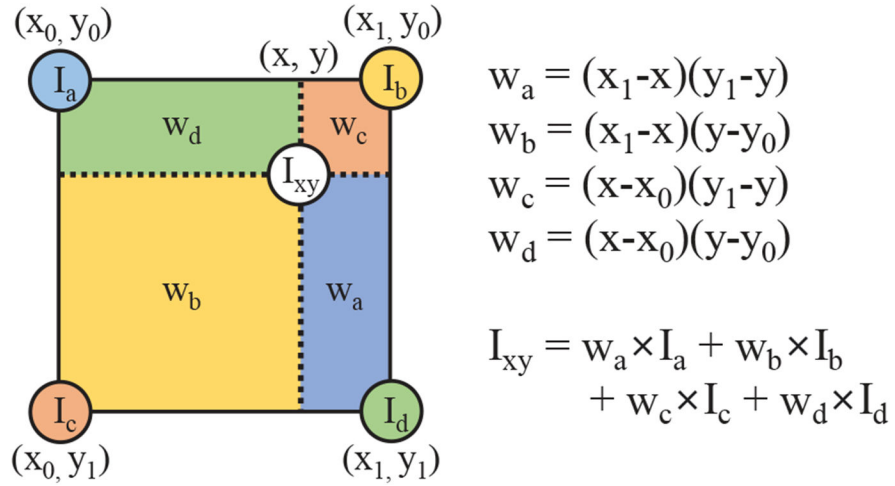$$+ w_c \times I_c + w_d \times I_d$$

Figure 3.
Bilinear Sampling of a pixel or feature at a coordinate (x,y) involves the bilinear interpolation of pixels or features at four adjacent points (I_a, I_b, I_c, I_d) in a grid by the weights (w_a, w_b, w_c, w_d).
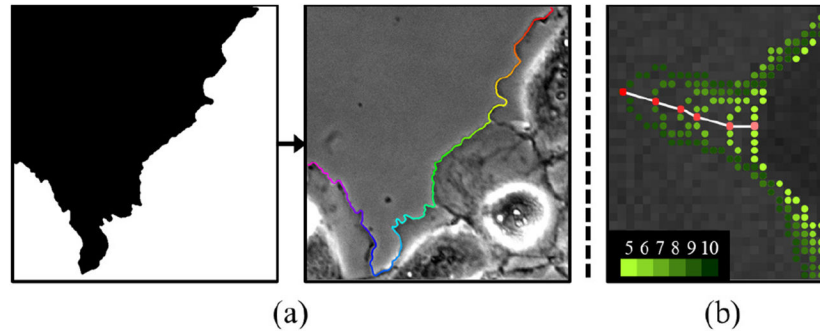
Figure 4.
(a) Extraction of contour points from the segmentation mask of one of the phase contrast live cell images [13]. Contour points are in sequential order as shown in color, from pink to red. **(b) Labeling tracking points** in 5x higher temporal resolution. The red point is tracked with correspondences shown in white lines. The color of the contour points changes from yellow-green to dark green as the frame number increases from 5 to 10.
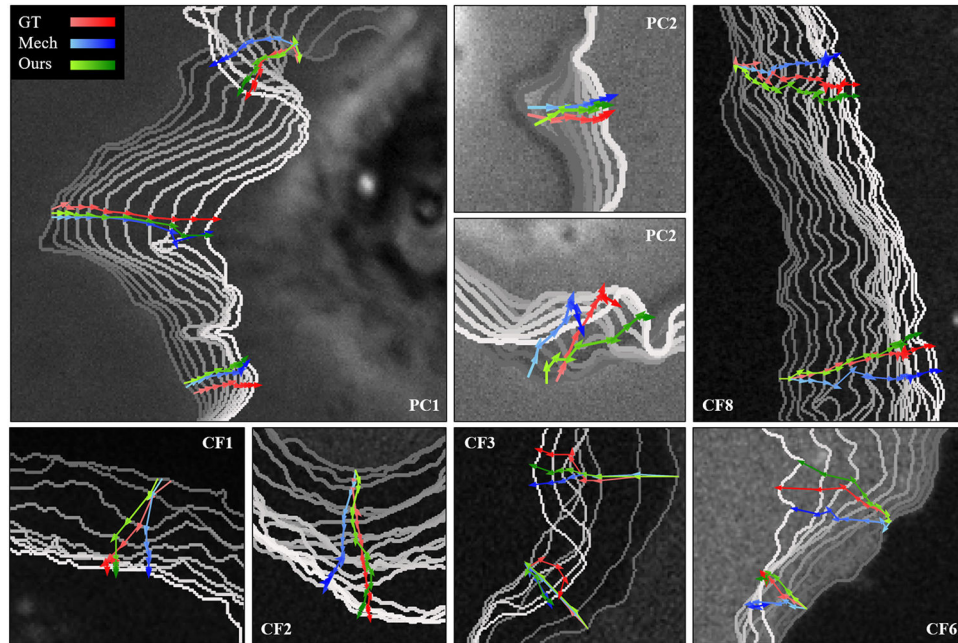
Figure 5. Trajectories of tracked points by our contour tracker (green), the mechanical model (blue), and ground truth labels (red).

To indicate the time from the initial frame to the last frame, the color of the trajectory gradually changes from light to dark, and the color of the contour changes from gray to white. PC prefix refers to phase contrast live cell dataset [13] and CF prefix refers to confocal fluorescence live cell dataset [36]. The number refers to the video number. Trajectories in PC1, PC2, and CF1 start from the mid-frame, and the rest of the trajectories start from the first frame at the same initial point.
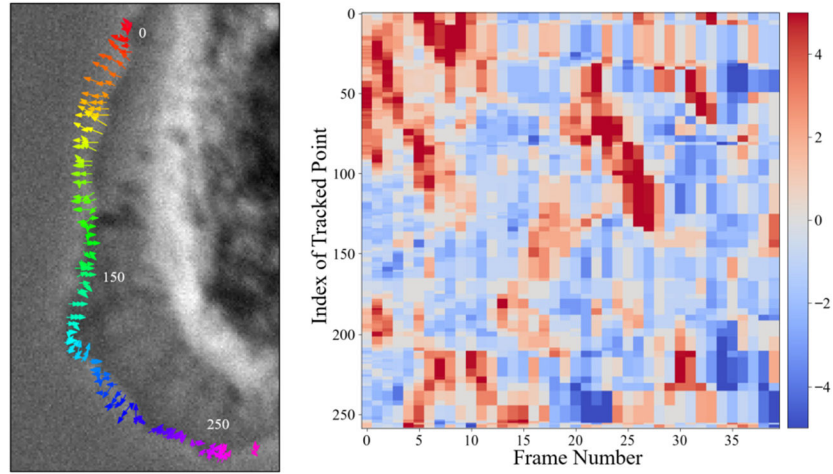
Figure 6. Quantification of morphodynamics from our contour tracking results.
Trajectories of tracked contour points for 3 frames on a phase contrast live cell [13] are shown on the left, and the quantification of those tracked points as velocities for 40 frames are shown on the right.
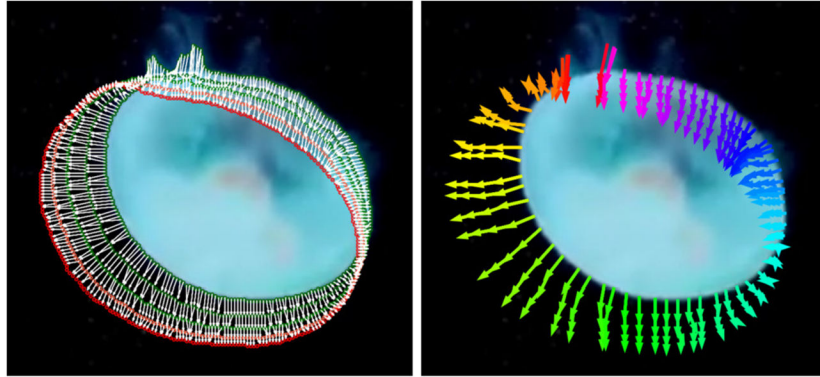
Figure 7. Visualization of contour tracking results on a jellyfish [30].
Dense correspondences between adjacent contours with white arrows are shown on the left.
The color of the contour changes from green to red as the frame number increases. The
trajectories of a few tracked points are shown on the right.

Table 1.

Ablation Studies of Loss functions on phase contrast live cell videos [13]. The Cycle refers to cycle consistency loss, Photo refers to photometric loss, Normal refers to mechanical-normal loss, and Linear refers to mechanical-linear loss.

Supervised	Cycle	Photo	Normal	Linear	SA.02	SA.04	SA.06	CA.01	CA.02	CA.03
✓					0.549	0.813	0.904	0.614	0.821	0.900
	✓				0.632	0.869	0.953	0.676	0.849	0.925
		✓			0.198	0.383	0.471	0.234	0.402	0.489
			✓		0.640	0.858	0.948	0.674	0.853	0.922
			✓	✓	0.378	0.593	0.686	0.400	0.594	0.674
	✓		✓	✓	0.426	0.611	0.738	0.469	0.627	0.710
	✓		✓	✓	**0.729**	**0.937**	**0.974**	**0.762**	**0.925**	**0.971**

Table 2.

Ablation studies of Architecture on phase contrast live cell videos [13].

Method	$SA_{.02}$	$SA_{.04}$	$SA_{.06}$	$CA_{.01}$	$CA_{.02}$	$CA_{.03}$
No Cross	0.659	0.858	0.969	0.696	0.851	0.939
Single Cross	0.677	0.864	0.930	0.734	0.854	0.913
Circ Conv	0.643	<u>0.931</u>	**0.983**	0.718	<u>0.910</u>	<u>0.965</u>
1D Conv	<u>0.692</u>	0.909	<u>0.976</u>	<u>0.736</u>	0.881	0.948
Ours	**0.729**	**0.937**	0.974	**0.762**	**0.925**	**0.971**

Table 3.

Comparison with other methods on phase contrast live cell videos [13].

Method	$SA_{.02}$	$SA_{.04}$	$SA_{.06}$	$CA_{.01}$	$CA_{.02}$	$CA_{.03}$
UFlow [14]	0.585	0.809	0.881	0.632	0.802	0.857
PoST [24]	0.629	0.850	<u>0.947</u>	0.693	<u>0.872</u>	<u>0.939</u>
Mechanical [21]	<u>0.683</u>	<u>0.853</u>	0.938	<u>0.722</u>	0.863	0.927
Ours	**0.729**	**0.937**	**0.974**	**0.762**	**0.925**	**0.971**

Table 4.

Comparison with other methods on confocal fluorescence live cell videos [36].

Method	$SA_{.02}$	$SA_{.04}$	$SA_{.06}$	$CA_{.01}$	$CA_{.02}$	$CA_{.03}$
UFlow [14]	0.605	0.785	0.863	0.520	0.685	0.791
PoST [24]	0.614	0.805	**0.888**	0.531	0.706	0.807
Mechanical [21]	0.603	0.798	0.876	0.517	0.711	0.804
Ours	**0.632**	**0.824**	0.882	**0.555**	**0.728**	**0.826**