Supplementary Material - 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

S.1. Addtional Methods

Mechanical-Linear Loss. Similar to linear spring force from the mechanical model [5], the mechanical-linear loss measures the average distance $A = \sum_{i=0}^{N-1} \|p_{t+1}^{i+1} - p_{t+1}^{i}\|_2$ between adjacent predicted contour points and forces every distance between adjacent points to be the same as the average distance. Here, each predicted contour point is $p_{t+1}^{i} = \phi(O_{t \to t+1}(p_t^i))$. The mechanical-linear is different from Snake [4] algorithm's tension term, which simply minimizes the distance between adjacent points.

$$L_{\text{mech-linear}} = \sum_{i=0}^{N_t-2} \| \|p_{t+1}^{i+1} - p_{t+1}^{i}\|_2 - A\|_1 \qquad (1)$$

Photometric Loss. Inspired by UFlow [3], we implemented photometric loss that minimizes the forward-then-backward tracked point's pixel intensity to match its original point's pixel intensity. Similar to the cycle consistency loss, forward and backward consistency losses are combined. But photometric loss measures the difference in pixel intensity, not the distance.

$$\begin{aligned} L_{photo} = &\sum_{i=0}^{N_t-1} \|\gamma(p_t^i) - \gamma(O_{t+1 \to t}(p_{t+1}^i))\|_1 \\ &+ \sum_{i=0}^{N_t-1} \|\gamma(p_{t+1}^i) - \gamma(O_{t \to t+1}(p_t^i))\|_1 \end{aligned} \qquad (2)$$

Retrieving a point's pixel intensity is denoted as γ.

Segmentation To provide an ordered sequence of contour points C_t as input to our contour tracker, the cell body is segmented first. Confocal fluorescence live cell videos [9] have a distinct boundary between the cell body and dark background, so conventional image thresholding is enough to obtain their segmentation masks. However, the live cell videos taken with phase contrast microscope [2] have challenging visual features such as halo and shade-off artifacts. As a result, a specialized deep segmentation model [2] for phase contrast live cell videos is adopted. Both datasets used in our paper are not manually segmented, so their segmentation masks contain some noise.

Labeling Tool We developed the labeling tool in Python to expedite the labeling process, as shown in Fig. S.1.1. It is available as an executable application in Windows 10/11. The labeling tool loads images with contour tracking points marked on the contour of the image. A user can click on the main view to create a tracking point, indicated by the blue circle. Scrolling up or down zooms in or out at the location of the user's cursor. The zoomed-in window is shown on the left pane of the GUI. Prev or next button on the bottom changes the frame such that the user can view how points move in consecutive frames and label tracking points. Created tracking points can be dragged to change their locations. Press the save button to save tracking points labeled at the current frame, or press the clear button to remove all labeled tracking points.

Training Details In this section, we provide training details about our contour tracker and two compared methods, UFlow [3], and PoST [6]. Our contour tracker, UFlow and PoST are all trained from scratch except for the ImageNet [1] pre-trained VGG16 encoder and ResNet50 encoder in our contour tracker and PoST, respectively. The architecture of UFlow [3] is kept the same, but we modified PoST [6] to improve its performance since their original architecture did not perform well in our dataset. Our contour tracker, UFlow [3], and PoST [6] are trained by unsupervised learning with cycle consistency loss. But our contour tracker is also trained with mechanical-normal loss. UFlow [3] only takes images as input, while PoST [6] and our contour tracker take both images and contour points as input for training and inference.

S.2. Addtional Results

We qualitatively validate dense point-to-point correspondences in the long sequences of phase contrast (PC) [2] and confocal fluorescence (CF) live cell videos [9] and a jelly-

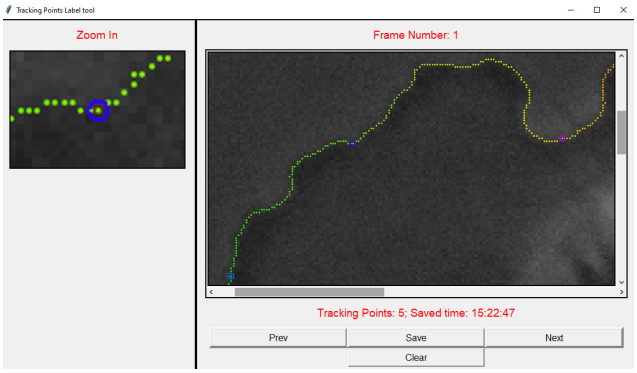

Figure S.1.1. GUI for labeling tracking points.

fish video [8] on our webpage at https://junbongjang.github.io/projects/contour-tracking/. They are best viewed in 1080p, HD resolution. The dense correspondences are shown with arrows pointing from the first frame's contour C_t to the second frame's contour C_{t+1}. The arrows are colored by the sequence of contour points' order from red to purple. The length of the arrow is the magnitude of the contour point's movement. The direction and magnitude of the arrows align well with our expectations, except for some abrupt changes in contour points due to the segmentation error.

For the jellyfish video, we picked 200 frames long sequence containing a single jellyfish from the Rainbow Jellyfish Benchmark [8]. Since our contour tracker is trained by unsupervised learning in less than a day, we train on a jellyfish video and predict dense point-to-point correspondences on the same video, which removes the need for making a training dataset.

Short & Long Term Tracking. In our main paper, we only evaluated long-term tracking of the points from the first to the last frame, so the point tracked from the previous frame was used to track in the next frames. Additionally, we evaluate both short-term and long-term tracking in Fig. S.2.1, which shows that the tracking error accumulates as the number of frames to track increases. At first, the Mechanical model [5] and our contour tracker has similar tracking accuracy, but the Mechanical model's [5] tracking accuracy decreases much faster than our contour tracker's. Cumulative Mean accuracy is obtained by averaging all SA or CA up to the frame number t, which is the x-axis in Fig. S.2.1.

Error Study. We performed an error study in the phase contrast videos [2]. The tracking accuracy is linearly proportional to the magnitude of the cellular movement. The spatial accuracy ($SA_{.02}$) decreases by about 0.02 as the absolute velocity increases by 1. The velocity is in unit pixels/frame and is perpendicular to the cellular contour. During cellular expansion and contraction, $SA_{.02}$ is 0.731 and

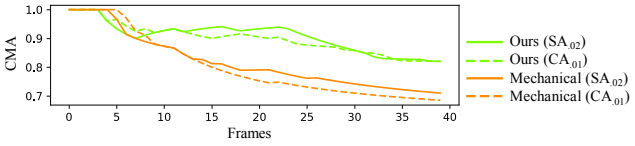

Figure S.2.1. Cumulative mean accuracy (CMA) from the first to the last frame on one of the confocal fluorescence videos [9].

0.710, respectively, so the direction of the cellular motion does not significantly affect the accuracy.

Forward and Backward Tracking. Only the forward cross attention is used to regress offsets for inference because forward tracking achieves significantly higher accuracy than backward tracking by 0.16/0.43 at $SA_{.02}$/$CA_{.01}$ in the phase contrast videos [2]. Ideally, both forward and backward tracking's performance should be similar, but our contour tracker is trained with mechanical-normal loss that updates the weights of the forward cross attention layer only during backpropagation. For computational efficiency during inference, the backward cross attention layer can be removed from our contour tracker such that backward offsets $O_{t+1 \to t}$ are not predicted.

S.3. Limitations

The segmentation error affects the performance of both the mechanical model and our contour tracker, which uses features from the contour and maps correspondence between contour points. For future work, jointly training our contour tracker with the segmentation model end-to-end will refine contours and then find contour point correspondences. Also, segmentation accuracy can be further improved by transfer learning with diverse microscopy datasets [2] or a human-in-the-loop approach without preparing large-scale datasets [7].

Our contour tracker can naturally handle many-to-one correspondences (merging) but not one-to-many correspondences (splitting). For future work, implementing one-to-many correspondences will handle expanding contours even better. One potential way is to regress backward offset from the second contour to the first contour and find many-to-one correspondences, which become one-to-many correspondences when reversed.

References

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1

[2] Junbong Jang, Chuangqi Wang, Xitong Zhang, Hee June Choi, Xiang Pan, Bolun Lin, Yudong Yu, Carly Whittle, Madison Ryan, Yenyu Chen, and Kwonmoo Lee. A deep learning-based segmentation pipeline for profiling cellular morphodynamics using multiple types of live cell microscopy. *Cell Reports Methods*, Oct 2021. 1, 2

[3] Rico Jonschkowski, Austin Stone, Jonathan T Barron, Ariel Gordon, Kurt Konolige, and Anelia Angelova. What matters in unsupervised optical flow. In *European Conference on Computer Vision*, pages 557–572. Springer, 2020. 1

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

[5] Matthias Machacek and Gaudenz Danuser. Morphodynamic profiling of protrusion phenotypes. *Biophysical journal*, 90(4):1439–1452, 2006. 1, 2

[6] Gunhee Nam, Miran Heo, Seoung Wug Oh, Joon-Young Lee, and Seon Joo Kim. Polygonal point set tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5569–5578, 2021. 1

[7] Marius Pachitariu and Carsen Stringer. Cellpose 2.0: how to train your own model. *Nature Methods*, pages 1–8, 2022. 2

[8] Ivan Skorokhodov, Sergey Tulyakov, and Mohamed Elhoseiny. 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. 2

[9] Chuangqi Wang, Hee June Choi, Sung-Jin Kim, Aesha Desai, Namgyu Lee, Dohoon Kim, Yongho Bae, and Kwonmoo Lee. Deconvolution of subcellular protrusion heterogeneity and the underlying actin regulator dynamics from live cell imaging. *Nature Communications*, 9(1):1688, Apr 2018. 1, 2