CTC: Cell Linking using Optical-Flow Enhanced Kalman Filtering

Raphael Reme, Tristan Manneville, Alasdair Newson, Elsa Angelini, Jean-Christophe Olivo-Marin, Thibault Lagache

I. Summary

We propose to use our Bayesian linking algorithms *KOFT* and *SKT*[1] that are implemented in our ByoTrack[2] Python library. They both rely on Kalman filters to model particle motion, and solve the tracks-to-detection association frame by frame with Jonker-Volgenant algorithm to find a solution to the Linear Association Problem (LAP). Whereas classical Bayesian approaches (like *SKT*) measure only the position (sometimes the intensity) of the tracked objects[3–8], in *KOFT* we use optical flow to also measure the velocity of these objects. More precisely, its Kalman filter is designed with a 2-steps update at time t: a first update is done with the position of the associated detection, a second update measures the *future* velocity of the track using optical flow between frame t and $t+1$ at the estimated localization of the track (see Figure 1). Cell mitosis events are detected through a second LAP at each frame between linked tracks and non-linked detections.

II. Method

In this section, we present our method called *Kalman and Optical Flow Tracking (KOFT)*. We also introduce a baseline method, referred to as *Standard Kalman Tracking (SKT)*, that follows closely *KOFT* implementation but does *not* exploit optical flow. In the following, each track i is modeled with an unobserved state $\mathbf{x}_t^i \in R^4$ which consists of positions and velocities. The detections $\mathbf{z}_t^i$ at frame t are the noisy measurements of the underlying tracks. In *KOFT* and *SKT*, the tracks' states are iteratively estimated from these measurements using Kalman filtering (see Figure 1).

A. Kalman filtering

Track state $\mathbf{x}_t^i$ is modeled as a Markov chain along time (process) and detections $\mathbf{z}_t^i$ are generated from the states (measurement) with the following model:

$$\mathbf{x}_t^i = \mathbf{F}\mathbf{x}_{t-1}^i + \mathbf{w}_t^i \qquad \text{[process]} \qquad (1)$$

$$\mathbf{z}_t^i = \mathbf{H}\mathbf{x}_t^i + \mathbf{v}_t^i, \qquad \text{[measurement]} \qquad (2)$$

This research is supported by the Institut Pasteur and France-BioImaging Infrastructure (ANR-10-INBS-04). R.R and T.L. are supported by the ANR (ANR-21-CE45-0020-01 REBIRTH).

Raphael Reme (raphael.reme@pasteur.fr), Tristan Manneville (tristan.manneville@pasteur.fr), Thibault Lagache (thibault.lagache@pasteur.fr) and Jean-Christophe Olivo-Marin (jcolivo@pasteur.fr) are with the BioImage Analysis Unit at Institut Pasteur, CNRS UMR 3691, as well as with Université Paris Cité in Paris, France.
Alasdair Newson (anewson@telcom-paris.fr) and Elsa Angelini (elsa.angelini@telecom-paris.fr) are with the LTCI unit at Telecom Paris of the Institut Polytechnique de Paris, France.

where $\mathbf{z}_t^i$ is the measurement vector of track i at time t, $\mathbf{F}$ is the process matrix and $\mathbf{H}$ is the measurement matrix. $\mathbf{w}_t^i$ and $\mathbf{v}_t^i$ are uncorrelated process and measurement noise vectors. They are modeled as zero-mean Gaussian noise vectors with $\mathbf{Q}$ and $\mathbf{R}$ as their covariance matrices.

Under these assumptions, Kalman filtering optimally and iteratively estimates the state distribution $(\mathbf{x}_t^i)_t$ from the observed measurements $(\mathbf{z}_t^i)_t$. Let $(\hat{\mathbf{x}}_{t-1}^i, \hat{\mathbf{P}}_{t-1}^i)$ be the mean and covariance of the state estimation at frame $t-1$ such that $\mathbf{x}_{t-1}^i|(\mathbf{z}_k^i)_{k<t} \sim \mathcal{N}(\hat{\mathbf{x}}_{t-1}^i, \hat{\mathbf{P}}_{t-1}^i)$. The estimation at frame t is computed in three steps (prediction, projection and update)[9,10]:

$$\bar{\mathbf{x}}_t^i = \mathbf{F}\hat{\mathbf{x}}_{t-1}^i, \qquad \bar{\mathbf{P}}_t^i = \mathbf{F}\hat{\mathbf{P}}_{t-1}^i\mathbf{F}^\top + \mathbf{Q} \quad \text{[prediction]} \qquad (3)$$

$$\mathbf{y}_t^i = \mathbf{z}_t^i - \mathbf{H}\bar{\mathbf{x}}_t^i, \qquad \mathbf{S}_t^i = \mathbf{H}\bar{\mathbf{P}}_t^i\mathbf{H}^\top + \mathbf{R} \quad \text{[projection]} \qquad (4)$$

$$\hat{\mathbf{x}}_t^i = \bar{\mathbf{x}}_t^i + \mathbf{K}_t^i\mathbf{y}_t^i, \quad \hat{\mathbf{P}}_t^i = \left(\mathbf{I} - \mathbf{K}_t^i\mathbf{H}\right)\bar{\mathbf{P}}_t^i, \quad \text{[update]} \qquad (5)$$

where $\mathbf{x}_t^i|(\mathbf{z}_k^i)_{k<t} \sim \mathcal{N}(\bar{\mathbf{x}}_t^i, \bar{\mathbf{P}}_t^i)$ is the prior (or predicted) state at time t, $(\mathbf{y}_t^i, \mathbf{S}_t^i)$ is the innovation and $\mathbf{K}_t^i = \bar{\mathbf{P}}_t^i\mathbf{H}^\top\mathbf{S}_t^{i^{-1}}$ the optimal Kalman gain.

B. Process model

Contrary to the original paper[1], in CTC datasets, cells usually follow Brownian-like motion. Therefore, we use a locally constant position model[5,6]. Tracks' states consist of positions and velocities: $\mathbf{x}_t^i = (x_t^i, \dot{x}_t^i, y_t^i, \dot{y}_t^i)$. Without any loss of generality, let $\mathrm{dt} = 1$ be the time frame interval:

$$\mathbf{F} = \begin{pmatrix} 1 & \mathrm{dt} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \mathrm{dt} \\ 0 & 0 & 0 & 0 \end{pmatrix}, \; \mathbf{Q} = \sigma_{\mathrm{acc}}^2 \begin{pmatrix} \frac{\mathrm{dt}^4}{4} & \frac{\mathrm{dt}^3}{2} & 0 & 0 \\ \frac{\mathrm{dt}^3}{2} & \mathrm{dt}^2 & 0 & 0 \\ 0 & 0 & \frac{\mathrm{dt}^4}{4} & \frac{\mathrm{dt}^3}{2} \\ 0 & 0 & \frac{\mathrm{dt}^3}{2} & \mathrm{dt}^2 \end{pmatrix} \qquad (6)$$

where σ_{acc} accounts for the expected velocity variations.

For each track state at frame $t-1$, we compute the prior state at frame t with our process model (Equation 3) before associating and updating with any measurement (see following sections).

C. Data association (linking)

At any time step t, our algorithm attempts to associate each predicted track $\bar{\mathbf{x}}_t^i$ with a detection $\mathbf{z}_t^j$ from frame t. Tracks that are successfully associated with a detection are said to be *linked*. To achieve this linking, we compute the Euclidean

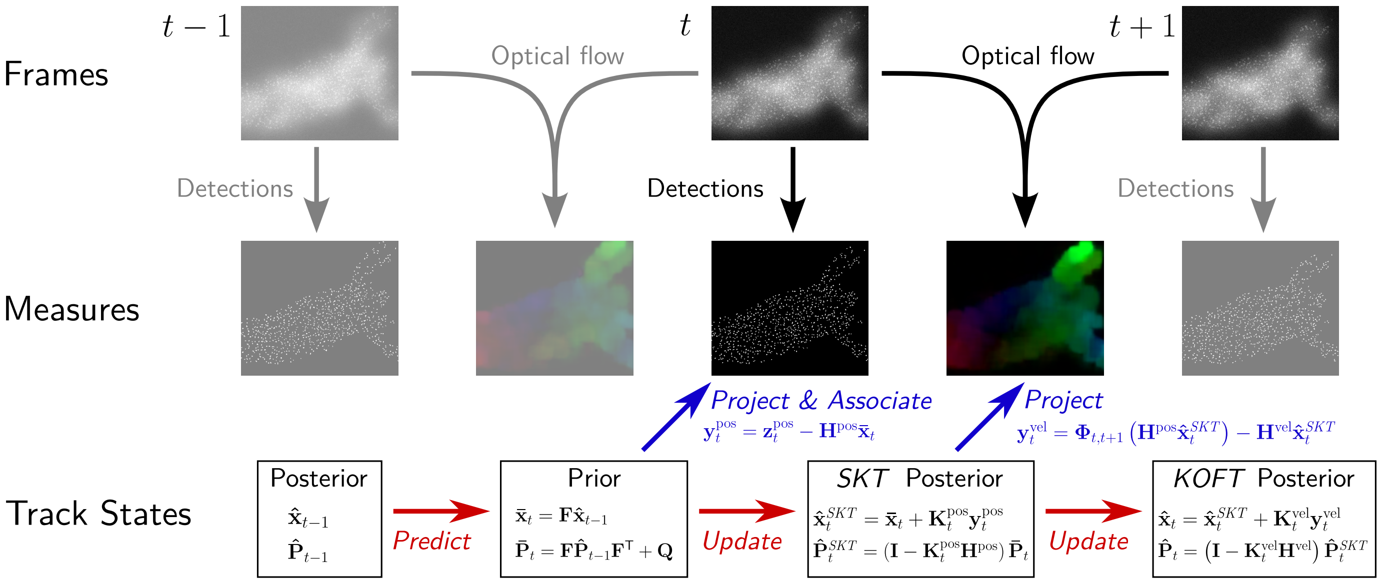

Fig. 1. **SKT & KOFT overview.** Kalman filtering is used to iteratively estimate track states (positions and velocities). In *SKT*, positions and velocities are updated from previous detections. In *KOFT*, an additional dense optical flow is computed between frame t and $t+1$ to measure the *forward* displacement of each pixel. This complementary information is integrated into the Kalman filter to improve the velocity estimates.

distance between predicted tracks and detections with $C_{ij} = \left|\left|\mathbf{H}\bar{\mathbf{x}}_t^i - \mathbf{z}_t^j\right|\right|$. Tracks-to-detections associations are found by minimizing the sum of distances using the Jonker-Volgenant algorithm[11]. We only considered associations below a fixed distance threshold η.

To detect splitting events (cell mitosis), we run a second linking step between linked tracks and non-linked detections. We bias association toward plausible splits with a modified cost that accounts for cell sizes. Let a track i of size ρ_i associated to a detection j of size ρ_j in the first step, the cost to associate i to another detection k of size ρ_k is defined as $\bar{C}_{ik} = C_{ik}\gamma_{jk}\gamma'_{ijk}$, where $\gamma_{jk} = \frac{\max(\rho_j,\rho_k)}{\min(\rho_j,\rho_k)}$ increases the cost when the two detections have different sizes and $\gamma'_{ijk} = \frac{\max(\rho_i,\rho_j+\rho_k)}{\min(\rho_i,\rho_j+\rho_k)}$ increases the cost when the children sizes do not sum to the parent track size.

D. Track creation & termination

As there are no false positive detections, new tracks are created from non-linked detections. To be robust to false detections, a new track is created only if its initial non-linked detection can be linked over N_{valid} frames.

To handle cases where an object may be difficult to detect for a short time, remaining non-linked tracks are maintained for N_{gap} frames. During this time, their states are *not* updated in the Kalman filter, but the prediction step is still carried out. After N_{gap} missed consecutive frames, we terminate the track.

E. Update states from detections

To update a track state from detections, we model the positional measurement noise as a Gaussian with zero mean and variance σ^2_{pos}. Our positional measurement model is therefore:

$$\mathbf{H}^{\text{pos}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \ \mathbf{R}^{\text{pos}} = \sigma^2_{\text{pos}}\mathbf{I}_2 \tag{7}$$

In both *SKT* and *KOFT*, the state of the track i linked to a detection $\mathbf{z}_t^j$ is updated according to Equations 4 and 5, where $\mathbf{H}$, $\mathbf{R}$ and $\mathbf{z}_t^i$ are given by $\mathbf{H}^{\text{pos}}$, $\mathbf{R}^{\text{pos}}$ and $\mathbf{z}_t^j$. We denote the resulting posterior state as $\hat{\mathbf{x}}_t^{\text{i},SKT}$: this is the posterior state used in *SKT*.

F. Update states from optical flow

In *KOFT*, we propose to measure objects velocity using optical flow and to further update the tracks state using this additional measure. *KOFT* is therefore designed with two update steps (Figure 1): for each frame, the track states are updated a first time with a positional measurement from detections (see Section II-E above), and then a second time, with a velocity measurement from the optical flow.

Let $\mathbf{\Phi}_{t,t+1}(\mathbf{z}) \in \mathbb{R}^2$ be the computed optical flow between frame t and $t+1$ at pixel position $\mathbf{z}$. To measure the velocity of a tracked object i at time t, the optical flow between frames t and $t+1$ (*i.e.* the *forward* displacements of the pixels) is first computed. Next, the velocity is extracted from the optical flow map at the object's expected position: $\mathbf{z}_t^{i,\text{vel}} = \mathbf{\Phi}_{t,t+1}(\mathbf{H}^{\text{pos}}\hat{\mathbf{x}}_t^{\text{i},SKT})$. We model the velocity measurement noise as Gaussian with zero mean and variance σ^2_{vel}. Our velocity measurement model is therefore:

$$\mathbf{H}^{\text{vel}} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \ \mathbf{R}^{\text{vel}} = \sigma^2_{\text{vel}}\mathbf{I}_2 \tag{8}$$

The *SKT* posterior state $\hat{\mathbf{x}}_t^{\text{i},SKT}$ is further updated according to equations 4 and 5, where $\mathbf{H}$, $\mathbf{R}$ and $\mathbf{z}_t^i$ are given by $\mathbf{H}^{\text{vel}}$, $\mathbf{R}^{\text{vel}}$ and $\mathbf{z}_t^{i,\text{vel}}$. We denote the resulting posterior as $\hat{\mathbf{x}}_t^i$.

G. Optical Flow

KOFT can be used with any robust optical flow algorithm. We decided to rely on Farneback algorithm[12]. We used the Open-CV implementation[13], where we only tune the window size parameter and kept the default values for all the other parameters.

H. Post-processing

We smooth the track positions after tracking. We assume a Gaussian positional noise and use the optimal Rauch–Tung–Striebel (RTS) smoother[14]. This slightly improves localization of tracks, but does not change any association.

III. Implementation details and parameters tuning

Implementations of *KOFT* and *SKT* are based on the Python library ByoTrack[2]. Data and code are available at https://github.com/raphaelreme/byotrack. All the Kalman filters in *SKT* and *KOFT* use the same hyper-parameters that are set using *ad-hoc* rules.

First, we compute three video-specific features from the provided cell segmentation: **(1)** the average cell radius ρ, **(2)** the average distance of the closest neighboring cell d_{closest} and **(3)** the increase α between the final and initial number of cells.

Then, we use these features to set the default values of our hyper-parameters:

- The window size w of Farneback is set with $w = \max(10, \frac{\rho}{2})$.
- The uncertainty of Kalman filters are set following $\sigma_{\text{pos}} = \frac{\rho}{2}$, $\sigma_{\text{vel}} = \sigma_{\text{acc}} = 3\rho$.
- The association threshold is fixed with $\eta = \max(3\rho, d_{\text{closest}})$.
- Handling splitting event is only done if $\alpha > 30\%$
- In the cell linking benchmark, there is no false positive detection, we therefore set $N_{\text{valid}} = 1$. There are few missed detections and we set by default $N_{\text{gap}} = 1$, allowing to bridge over 1 missed detection.
- For 3D videos, we did not find a robust optical flow and decided to simply use *SKT* which is faster and already yields good performance.

References

[1] R. Reme, A. Newson, E. Angelini, J.-C. Olivo-Marin, and T. Lagache, “Particle tracking in biological images with optical-flow enhanced kalman filtering,” in *2024 IEEE International Symposium on Biomedical Imaging (ISBI)*. IEEE, 2024, pp. 1–5.

[2] A. Hanson, R. Reme, N. Telerman, W. Yamamoto, J.-C. Olivo-Marin, T. Lagache, and R. Yuste, “Automatic monitoring of neural activity with single-cell resolution in behaving hydra,” *Scientific Reports*, vol. 14, no. 1, p. 5083, 2024.

[3] J.-Y. Tinevez, N. Perry, J. Schindelin, G. M. Hoopes, G. D. Reynolds, E. Laplantine, S. Y. Bednarek, S. L. Shorte, and K. W. Eliceiri, “Trackmate: An open and extensible platform for single-particle tracking,” *Methods*, vol. 115, pp. 80–90, 2017.

[4] I. Smal, K. Draegestein, N. Galjart, W. Niessen, and E. Meijering, “Particle filtering for multiple object tracking in dynamic fluorescence microscopy images: Application to microtubule growth analysis,” *IEEE Transactions on Medical Imaging*, vol. 27, no. 6, pp. 789–804, 2008.

[5] A. Genovesio, T. Liedl, V. Emiliani, W. Parak, M. Coppey-Moisan, and J.-C. Olivo-Marin, “Multiple particle tracking in 3-d+t microscopy: method and application to the tracking of endocytosed quantum dots,” *IEEE Transactions on Image Processing*, vol. 15, no. 5, pp. 1062–1070, 2006.

[6] N. Chenouard, I. Bloch, and J.-C. Olivo-Marin, “Multiple hypothesis tracking for cluttered biological image sequences,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2736–3750, 2013.

[7] K. Ulicna, G. Vallardi, G. Charras, and A. R. Lowe, “Automated deep lineage tree analysis using a bayesian single cell tracking approach,” *Frontiers in Computer Science*, vol. 3, p. 734559, 2021.

[8] L. Feng, Y. Xu, Y. Yang, and X. Zheng, “Multiple dense particle tracking in fluorescence microscopy images based on multidimensional assignment,” *Journal of structural biology*, vol. 173, no. 2, pp. 219–228, 2011.

[9] R. E. Kalman, “A New Approach to Linear Filtering and Prediction Problems,” *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 03 1960. [Online]. Available: https://doi.org/10.1115/1.3662552

[10] G. Bishop, G. Welch *et al.*, “An introduction to the kalman filter,” *Proc of SIGGRAPH, Course*, vol. 8, no. 27599-23175, p. 41, 2001.

[11] R. Jonker and A. Volgenant, “A shortest augmenting path algorithm for dense and sparse linear assignment problems,” *Computing*, vol. 38, no. 4, pp. 325–340, 1987.

[12] G. Farnebäck, “Two-frame motion estimation based on polynomial expansion,” in *Image Analysis: 13th Scandinavian Conference, SCIA*. Springer, 2003, pp. 363–370.

[13] G. Bradski, “The OpenCV Library,” *Dr. Dobb’s Journal of Software Tools*, 2000.

[14] H. E. Rauch, F. Tung, and C. T. Striebel, “Maximum likelihood estimates of linear dynamic systems,” *AIAA journal*, vol. 3, no. 8, pp. 1445–1450, 1965.