PARTICLE TRACKING IN BIOLOGICAL IMAGES WITH OPTICAL-FLOW ENHANCED KALMAN FILTERING

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

PARTICLE TRACKING IN BIOLOGICAL IMAGES WITH OPTICAL-FLOW ENHANCED KALMAN FILTERING

Raphael Reme[⋆†] *Alasdair Newson*[†] *Elsa Angelini*[†]
Jean-Christophe Olivo-Marin[⋆] *Thibault Lagache*[⋆]

[⋆] Institut Pasteur, Université de Paris, CNRS UMR 3691, BioImage Analysis Unit F-75015 Paris, France
[†] LTCI, Telecom Paris, Institut Polytechnique de Paris, France
Corresponding author: raphael.reme@pasteur.fr

ABSTRACT

Single-particle-tracking is a fundamental prerequisite for studying biological processes in time-lapse microscopy. However, it remains a challenging task in many applications where numerous particles are driven by fast and complex motion patterns. To anticipate the motion of particles most tracking algorithms usually assume near-constant position, velocity or acceleration over consecutive frames. However, such assumptions are not robust to large and sudden velocity changes that typically occur in *in vivo* imaging. In this paper, we exploit optical flow to directly measure the velocity of particles in a Kalman filtering context. The resulting method shows improved robustness to correctly predict particles positions, even with sudden motions. We validate our method on simulated images with high particle density and fast elastic motion patterns. Quantitative results show a decrease of tracking errors by a factor of two, when compared to other tracking algorithms, while preserving fast computational time.

***Index Terms*—** Single-Particle-Tracking, Optical Flow, Kalman Filtering

1. INTRODUCTION

Single-particle-tracking (SPT) consists in reconstructing the trajectories of biological particles (e.g. molecules, pathogens or cells) from their time-lapse imaging, and studying their dynamics to gain information about biological processes. SPT algorithms are usually divided into two different steps: First, particle detection is performed on each time frame. Then, detections are linked into coherent tracks.

Sophisticated tracking algorithms have been developed over the years to robustly link detections in cluttered environment, i.e. in situations with high densities of particles and poor signal-to-noise ratio (SNR) leading to missed and false detections in each time frame. A first class of algorithms relies on global distance minimization (GDM) between detections. Since solving the GDM problem over all frames is infeasible due to memory and time limitations, heuristic methods are used. For example, some methods use post-processing (e.g. merging, splitting) [1, 2, 3, 4], after an initial frame-to-frame association of detections into tracks. Tracking can also be solved with probabilistic frameworks, using Kalman or particle filters with likelihood maximization [5, 6]. Multiple Hypothesis Tracking (MHT) can help to solve the association problem. This keeps several hypotheses at each frame, allowing the final linking decision to be made after seeing a few subsequent frames [7].

Most of these methods embed motion models generally designed with Kalman filters, which have proved to be highly effective for tracking particles in biology. However, Kalman filters have three major limitations when applied to *in vivo* imaging. First, since these models generally assume near-constant velocity or acceleration between three consecutive frames, they cannot handle fast and large motions that typically occur in living animals due to breathing and muscle contractions. Second, Kalman filtering assumes that the motion of each particle (e.g. cell) is independent, which is not the case for particles that are embedded in a deformable medium such a tissue. Third, the predicted position at the current frame only exploits information from the previous frames, which is not robust to sudden change of motion amplitude or direction.

In the case of only few particles, optical flow framework has proven to be robust to *in vivo* imaging conditions, by linking detections via recursively computing optical flows to correct motions between frames [8, 9, 10, 11, 12].

In natural videos, deep learning has become the gold standard to identify objects (e.g. humans, cars, etc.) between frames. These approaches require annotated data, and struggle to track biological particles as they lack discriminating features such as hair, clothing, color, etc. [13, 14, 15]

To robustly track numerous biological particles *in vivo*, we propose a *Kalman and Optical Flow Tracking (KOFT)*, that directly exploits optical flow within a Kalman filtering framework. To the best of our knowledge, this idea was experimented only once on natural images in DKFlow [16], leading to small tracking improvements. Our method implements Kalman filtering with two update steps: The first step integrates detection position information into track states. The second integrates velocity measured by optical flow at the track positions. Optical flow is computed with the next frame for robustness against fast and sudden motions. Moreover, thanks to optical flow regularization, velocities are consistent between spatially-close particles.

We demonstrate the performance of our method on realistic synthetic data designed to mimic fluorescence images of cells within a freely-behaving animal with complex elastic motions. We show that *KOFT* outperforms other approaches, making at least twice fewer errors while keeping a low computational cost.

2. METHOD

In this section, we detail the different steps in *Kalman and Optical Flow Tracking (KOFT)*. We also introduce a baseline method, referred to as *Simple Kalman Tracking (SKT)*, that do *not* exploit optical flow. In the following, each track i is modeled with an unobserved state $\mathbf{x}_t^i$ which consist of positions and velocities. The detections $(\mathbf{z}_t^j)_j$ at each frame t are the noisy measurements of the underlying tracks. In *KOFT* and *SKT*, the tracks' states are estimated from these measurements using Kalman filtering. (Figure 1).

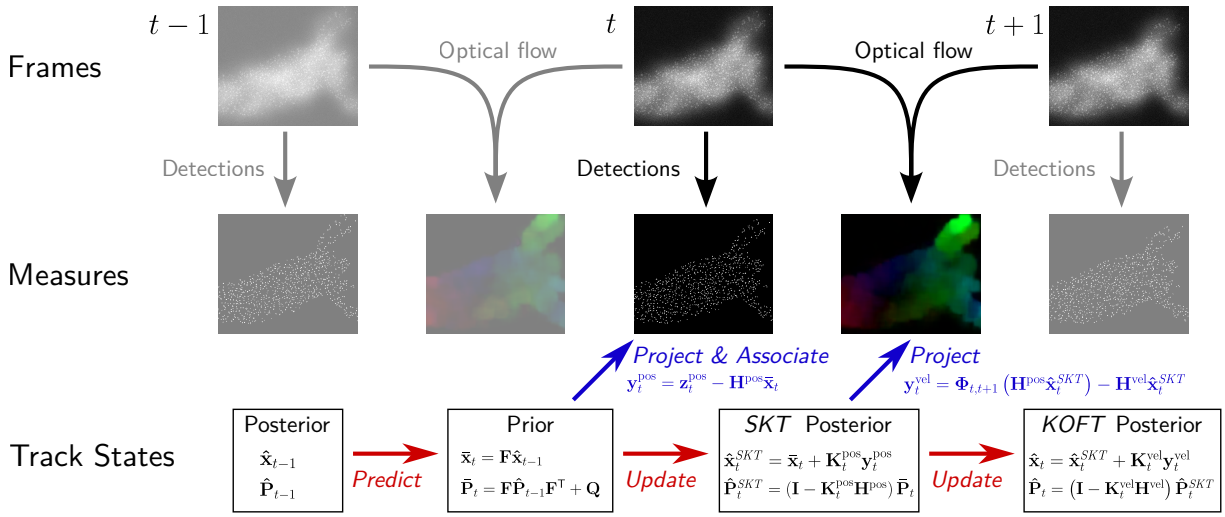

Fig. 1. **SKT & KOFT overview.** Kalman filtering is used to iteratively estimate track states (positions and velocities). In *SKT*, positions and velocities are updated from previous detections. In *KOFT*, an additional dense optical flow is computed between frame t and $t+1$ to measure the *future*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

2.1. Kalman filtering

Each track i is modeled with a Gaussian random state $\mathbf{x}_t^i$ which is updated through time and from which measurements are generated following

$$\mathbf{x}_t^i = \mathbf{F}\mathbf{x}_{t-1}^i + \mathbf{w}_t^i \tag{1}$$

$$\mathbf{z}_t^i = \mathbf{H}\mathbf{x}_t^i + \mathbf{v}_t^i, \tag{2}$$

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 system state 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 $t-1$. The estimation at frame t is computed in three steps (prediction, projection, update):

$$\bar{\mathbf{x}}_t^i = \mathbf{F}\hat{\mathbf{x}}_{t-1}^i, \qquad \bar{\mathbf{P}}_t^i = \mathbf{F}\hat{\mathbf{P}}_{t-1}^i\mathbf{F}^\top + \mathbf{Q} \tag{3}$$

$$\mathbf{y}_t^i = \mathbf{z}_t^i - \mathbf{H}\bar{\mathbf{x}}_t^i, \qquad \mathbf{S}_t^i = \mathbf{H}\bar{\mathbf{P}}_t^i\mathbf{H}^\top + \mathbf{R} \tag{4}$$

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

where $(\bar{\mathbf{x}}_t^i, \bar{\mathbf{P}}_t^i)$ 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.

In traditional Kalman tracking, measurements consist of the tracks' positions. We extend this framework, by additionally measuring the velocity of tracks thanks to optical flow.

2.2. Process model

We use a constant velocity model (also used in [5, 7]), which assumes that the velocities of particles are constant between two consecutive frames. Tracks' states consist of positions and velocities: $\mathbf{x}_t^i = (x_t^i, \dot{x}_t^i, y_t^i, \dot{y}_t^i)$. Let $\text{dt} = 1$ be the time frame interval. We assume a piece-wise constant Gaussian white noise acceleration model, with $\sigma_{\text{acc}} = 1.5$ (see [17] for details). The process is modeled with:

$$\mathbf{F} = \begin{pmatrix} 1 & \text{dt} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \text{dt} \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{Q} = \sigma_{\text{acc}}^2 \begin{pmatrix} \frac{\text{dt}^4}{4} & \frac{\text{dt}^3}{2} & 0 & 0 \\ \frac{\text{dt}^3}{2} & \text{dt}^2 & 0 & 0 \\ 0 & 0 & \frac{\text{dt}^4}{4} & \frac{\text{dt}^3}{2} \\ 0 & 0 & \frac{\text{dt}^3}{2} & \text{dt}^2 \end{pmatrix} \tag{6}$$

2.3. Data association (linking)

At any time step t, our algorithm attempts to associate each track with a detection in frame t. Tracks that are successfully associated with a detection are said to be *linked*. To achieve this linking, we compute the likelihood that a detection $\mathbf{z}_t^j$ is from track i as: $\Lambda_t^i(\mathbf{z}_t^j) = \mathcal{N}(\mathbf{z}_t^j;\ \mathbf{H}\bar{\mathbf{x}}_t^i, \mathbf{S}_t^i)$. Associations of tracks with detections are chosen to maximize the sum of log likelihoods using the Jonker-Volgenant algorithm [18]. We only consider associations with likelihoods above a certain threshold η.

2.4. Track creation & termination

After the linking process, there are two types of detections: those which are linked with a track, and those which are non-linked. New tracks are potentially created from the non-linked detections. To be robust to false detections, a new track is created only if a non-linked detection can be followed over $N_{\text{valid}} = 3$ frames.

To handle cases where a particle may be difficult to detect for a short time, remaining non-linked tracks are maintained for $N_{\text{gap}} = 7$ 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. Allowing linking a track after too many consecutive missed detections is counterproductive, because the uncertainty of its Kalman filter increases. This will potentially result in the track being associated with a false detection.

2.5. Update from detections

A linked track i is associated with a detection characterized by its position coordinates: $\mathbf{z}_t^{j,\text{pos}} = (x_t^j, y_t^j)$. The measurements noise is modeled as a zero-mean Gaussian noise with $\sigma_{\text{pos}} = 2$ pixels. Our positional measurement model is:

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

In both *SKT* and *KOFT*, the state of the track i 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 $\mathbf{z}_t^{i,\text{pos}}$. We denote the resulting *SKT* posterior state as $\hat{\mathbf{x}}_t^{i,SKT}$.

2.6. Update from optical flow

In *SKT*, the track velocity at time t is estimated only from its previous measured positions. This estimation is accurate only for motions that are slow with respect to the frame rate.

To improve robustness to fast and sudden motions, we propose to use optical flow to measure the velocities of particles. Let $\mathbf{\Phi}_{t,t+1}(\mathbf{z}) \in \mathbb{R}^2$ be the optical flow between frame t and $t+1$ at pixel position $\mathbf{z}$. The information provided by dense optical flow is based on pixel intensity and is often regularized globally in the image. Thus, optical flow provides useful, complementary information that can be integrated in the Kalman filter. In *KOFT*, the state at frame t is updated using the optical flow between frame t and $t+1$ (i.e. future displacements). The benefits are two-folds: (1) it foresees sudden changes from frame t to $t+1$ when updating state at time t, (2) it uses regularized velocity estimates, more consistent between spatially-close particles.

Unlike [16], we design a Kalman Filter with two update steps (Figure 1): First with positional measurements, then with velocity measurements. Thanks to this 2-steps procedure, posterior positions can be used, rather than prior/measured ones, to extract velocities from the optical flow map. Moreover, this formulation facilitates the update of velocities for all tracks, including non-linked ones, reducing uncertainty and improving motion estimation when particles are temporarily undetected. Our algorithm is therefore more robust to false positive association when the particle is undetected.

We tested two variants of *KOFT*: (1) *KOFT*--: updating position and velocity in a single update as in [16], (2) *KOFT*: use two separate updates, allowing to update velocity for non-linked tracks.

Optical flow method. We have compared Farneback [19], TVL1 [20, 21] and deep-learning based RAFT [22] algorithms on a video of *Hydra Vulgaris* neurons [23]. All images are first smoothed via Gaussian filtering ($\sigma = 1$ pixels) and downscaled four times. This preprocessing is done to speed up computations, improve noise robustness and extract global, meaningful motions. A quantitative substudy not reported here indicated that Farneback was faster and more accurate to estimate flow on our data. We emphasize that *KOFT* can be used with any performing optical flow method.

Update from velocity measurements. For each track i at time t (linked or non-linked), we extract measured velocities from the optical flow map $\mathbf{z}_t^{i,\text{vel}} = \mathbf{\Phi}_{t,t+1}(\mathbf{H}^{\text{pos}}\hat{\mathbf{x}}_t^{i,SKT})$. The velocity measurement noise is modeled as a zero-mean Gaussian noise with $\sigma_{\text{vel}} = 2$. Our velocity measurement model is:

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

The *SKT* posterior state $\hat{\mathbf{x}}_t^{i,SKT}$ is further updated with velocities 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 $\hat{\mathbf{x}}_t^i$.

3. SYNTHETIC IMAGE SIMULATOR

We evaluate our tracking performances on synthetic simulated data. Indeed, there is no annotated data available and the labeling task is labor intensive and prone to human errors and biases [24, 25] when dealing with fluorescence microscopy.

Existing simulators for particle tracking [24, 5, 2, 6, 25] rely on simple motion models (e.g. independent particles with Brownian or partially directed motion) and simple appearance models for particles (e.g. Gaussian profiles with Gaussian or Poisson noise, without background). These are not representative of the complex motion patterns targeted by our method. We have therefore designed our own simulator to simulate challenging tracking ground truths for particles moving within a highly deformable background (typically cells in tissue or freely-behaving animal).

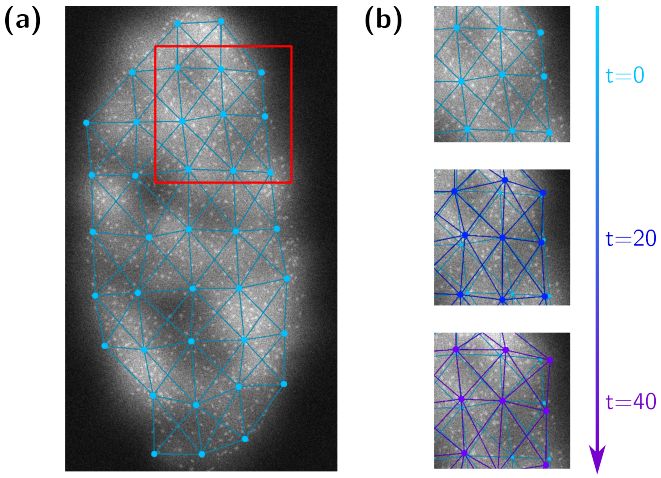

Fig. 2. **Synthetic image simulator (spring-based). (a)**: Visual appearance of the initial frame of the synthetic movie with springs (blue) between particles to track. **(b)**: Motion induced by random forces and springs constraints over time.

First, we model the animal's body or tissue as a global background signal, and position an initial population of particles within this body. Then, we apply global, elastic motions and deformation to particles and background using a spring network (Figure 2) or optical flow.

3.1. Particle, background & noise modeling

A particle is characterized by its shape, intensity and position, which all change in time. Particle shape is modeled as a 2D Gaussian profile. The covariance of the Gaussian profile is random (1 to 3 pixels wide). Positions of the initial population of particles are drawn from a Uniform distribution inside the body, with a control on the minimum distance between particles. The particles' signal is linearly mixed with a background signal. The background is modeled as a mixture of Gaussian profiles with large covariances (20 to 60 pixels wide) within the body. An additional Poisson Shot Noise is used to model fluorescence microscopy acquisition noise.

3.2. Motion modeling

At each time-step, we update the position of both background and particles using the same global motion. We propose two plausible global elastic motion models:

Optical flow motion. We extract optical flows from consecutive frames of a real video (in this work, we use a fluorescence video of *Hydra Vulgaris* neurons from [23]). These flows are applied to update the positions of both particles and background.

Springs motion. We model the animal's body or tissue with n points of equal mass attached by springs. (See figure 2). We apply random forces to the system and solve the n-body damped harmonic oscillator equations. Let $(\mathbf{p}_i \in \mathbb{R}^2)_{1 \leq i \leq n}$ be the coordinates of the n mass-points of our system. For each $\mathbf{p}_i$, we define its dampening coefficient $\lambda_i \in \mathbb{R}_+$. For each pair $(\mathbf{p}_i, \mathbf{p}_j)$, a spring is created with a stiffness $k_{ij} \in \mathbb{R}_+$ and an equilibrium length $l_{ij} \in \mathbb{R}_+$. We set

Motion	Springs			Optical Flow			FPS
Detections	Fake@90%	Fake@70%	Wavelet [26]	Fake@90%	Fake@70%	Wavelet [26]	
u-track (Fiji)[2, 27, 28]	$85.7 \pm 4.9\%$	$50.9 \pm 5.8\%$	$79.2 \pm 5.2\%$	$82.2 \pm 0.5\%$	$55.1 \pm 0.7\%$	$67.2 \pm 1.2\%$	25
eMHT (Icy)[7, 29]	$87.0 \pm 5.3\%$	$58.1 \pm 8.4\%$	$83.2 \pm 3.7\%$	$77.0 \pm 0.4\%$	$56.7 \pm 0.6\%$	$70.9 \pm 0.9\%$	4
SKT (ours)	$84.6 \pm 4.4\%$	$44.9 \pm 4.8\%$	$80.1 \pm 4.2\%$	$82.6 \pm 0.7\%$	$44.1 \pm 0.9\%$	$72.1 \pm 1.0\%$	25
KOFT-- (ours)	$95.5 \pm 1.3\%$	$71.4 \pm 5.2\%$	$89.5 \pm 2.8\%$	$94.6 \pm 0.5\%$	$69.7 \pm 1.0\%$	$84.6 \pm 0.5\%$	10
KOFT (ours)	$\mathbf{97.4} \pm 0.7\%$	$\mathbf{88.6} \pm 2.3\%$	$\mathbf{91.8} \pm 2.5\%$	$\mathbf{96.6} \pm 0.4\%$	$\mathbf{89.6} \pm 0.7\%$	$\mathbf{86.8} \pm 0.4\%$	10

Table 1. HOTA at 2 pixels [30] of the different tracking algorithms, on six synthetic scenarios (2 different motion patterns and three detection methods). We report the mean and std on 5 different random simulation sharing the same parameters (except the random seed). Frame per second (FPS) are computed with Fake@90% on the same laptop.

$k_{ij} = 0$ (no spring) for all pairs but the 8 closest neighbors. This system follows:

$$\ddot{\mathbf{p}}_i = \mathbf{f}_i(t) - \lambda_i \dot{\mathbf{p}}_i - \sum_j k_{ij}(||\mathbf{p}_i - \mathbf{p}_j||_2 - l_{ij}) \frac{\mathbf{p}_i - \mathbf{p}_j}{||\mathbf{p}_i - \mathbf{p}_j||_2} \quad (9)$$

with $\mathbf{f}_i(t) \in \mathbb{R}^2$ some random noise applied to $\mathbf{p}_i$ at time t.

Particles and background motions are computed through an elastic interpolation (Thin Plate Spline) between the n mass-points.

4. RESULTS

We validate our method on our synthetic data using optical flow motion or spring-based motion with 200 frames (1000x1000 pixels) and 1000 particles. For optical flow motions, we selected 200 frames from a fluorescence video of *Hydra Vulgaris* neurons [23], where the animal is contracting. The implementation of *KOFT* is based on the python library ByoTrack [31] and data and code are available at `https://github.com/raphaelreme/koft`.

To detect particles, we used the wavelet thresholding method described in [26] for which we measured a performance of around 85% recall and precision in our synthetic dataset. To isolate the impact of the detection method, we also implemented an artificial ("fake") detector, which uses the ground truth to generate detections with a known $\sigma_{\text{pos}} = 0.5$ pixels and fixed false negative (fnr) and false positive rates (fpr). We denote by Fake@f1, the fake detector with fpr = fnr = $1 - \text{f1}$.

In optical flow simulation, the animal is strongly contracting. Thus, particles tend to be closer, leading to poorer Wavelet detection performances (f1 score around 80% versus 85% in spring-based simulation). Tracking is consequently harder in this scenario.

We measure the HOTA [30] scores at a matching distance of 2 pixels and compare *SKT*, *KOFT*, *eMHT*[7] from Icy software [29] and *u-track* [2] from trackmate/Fiji software [27, 28]. *eMHT* is a probabilistic algorithm based on multiple motion models, with a probabilistic handling of tracks and a multiple hypothesis association method. With our density of particles, we were only able to run the multiple hypothesis association with a tree depth of 2 frames. *u-track* is a global distance minimization algorithm in two steps (frame-to-frame linking, tracklet stitching). We use the advanced version that model motion with Kalman filters. [27, 28]

All these algorithms are parameterized by an association cost limit η to prevent linking a non-detected track with an unlikely false detection. For each method, we report the metrics with the best threshold η found by grid search.

Table 1 summarizes the different performances of the algorithms. *u-track* [2, 27, 28] shares the same motion model than *SKT* and differs only in the tracks' creation and termination. It yields very similar results than *SKT*. Both are outperformed by *eMHT* [7, 29], in particular in a poor detections quality context.

Our new algorithm *KOFT*, is more robust to fast motions, enabling impressive performance with few false or missed detections (HOTA above 95%). Even with poor detection quality, it still reaches great performance (HOTA above 85%), outperforming other algorithms and dividing tracking errors by at least two (e.g. from 29.1% of errors for *eMHT* to only 13.2% for *KOFT* on flow-based simulation with Wavelet detections).

KOFT-- suffers from its single update Kalman filter, which does not update non-linked tracks' velocities. On the other hand, *KOFT* successfully exploits the additional information concerning tracks' motion to wait longer for lost tracks, significantly improving the results in high fnr contexts.

Dense optical flow is computationally expensive, but with our preprocessing and Farneback [19] method from opencv [32], it runs at 25 FPS. Therefore, *KOFT* is twice as fast (10 FPS) as *eMHT*, because it uses a much simpler association method.

5. CONCLUSION

In this paper, we proposed a novel method to improve single-particle-tracking in high density and real motions scenarios. Our algorithm exploits both optical flow and Kalman filtering to precisely predict particles motion.

Optical flow provides complementary information on the particles velocities, and is particularly helpful for robustness to sudden motions and to continue the correct tracking of undetected particles over a few frames. Our association and track management methods are simple but effective. We believe that more complex approaches could improve performances. For instance, we could employ multiple hypothesis or deep-learning based association.

We have simulated challenging realistic tracking data, with a high particle density and fast, elastic motions. Our method drastically improves tracking performances in these scenarios, dividing tracking errors by a factor of two.

6. COMPLIANCE WITH ETHICAL STANDARDS

This is a numerical simulation study for which no ethical approval was required.

7. ACKNOWLEDGMENTS

This work 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).

None of the authors declare to have a financial conflict of interest in the results of this study

8. REFERENCES

[1] I. F. Sbalzarini and P. Koumoutsakos, "Feature point tracking and trajectory analysis for video imaging in cell biology," *Journal of structural biology*, vol. 151, no. 2, pp. 182–195, 2005.

[2] K. Jaqaman, D. Loerke, M. Mettlen, et al., "Robust single-particle tracking in live-cell time-lapse sequences," *Nature methods*, vol. 5, no. 8, pp. 695–702, 2008.

[3] T. Lagache, A. Hanson, J. Pérez-Ortega, et al., "Tracking calcium dynamics from individual neurons in behaving animals," *PLoS computational biology*, vol. 17, pp. e1009432, 10 2021.

[4] R. Reme, V. Piriou, A. Hanson, et al., "Tracking intermittent particles with self-learned visual features," in *IEEE International Symposium on Biomedical Imaging (ISBI)*. IEEE, 2023, pp. 1–5.

[5] A. Genovesio, T. Liedl, V. Emiliani, et al., "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] I. Smal, K. Draegestein, N. Galjart, et al., "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.

[7] 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.

[8] T. Senst, V. Eiselein, and T. Sikora, "Robust local optical flow for feature tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 9, pp. 1377–1387, 2012.

[9] D. Guo, A. L. Van de Ven, and X. Zhou, "Tracking and measurement of the motion of blood cells using optical flow methods," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 3, pp. 991, 2014.

[10] T. Yamane, Y. Shirai, and J. Miura, "Person tracking by integrating optical flow and," *Biological Cybernetics*, vol. 63, pp. 421–431, 1990.

[11] A. Hand, T. Sun, D. Barber, et al., "Automated tracking of migrating cells in phase-contrast video microscopy sequences using image registration," *Journal of microscopy*, vol. 234, no. 1, pp. 62–79, 2009.

[12] S. Yang, D. Kohler, K. Teller, et al., "Nonrigid registration of 3-d multichannel microscopy images of cell nuclei," *IEEE Transactions on Image Processing*, vol. 17, no. 4, pp. 493–499, 2008.

[13] Y. Du, Z. Zhao, Y. Song, et al., "Strongsort: Make deepsort great again," *IEEE Transactions on Multimedia*, 2023.

[14] Y. Zhang, T. Wang, and X. Zhang, "Motrv2: Bootstrapping end-to-end multi-object tracking by pretrained object detectors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 22056–22065.

[15] Y. Yao, I. Smal, I. Grigoriev, et al., "Deep-learning method for data association in particle tracking," *Bioinformatics*, vol. 36, no. 19, pp. 4935–4941, 07 2020.

[16] Y. Chen, D. Zhao, and H. Li, "Deep kalman filter with optical flow for multiple object tracking," in *IEEE international conference on systems, man and cybernetics (SMC)*, 2019, pp. 3036–3041.

[17] R. Labbe, "Kalman and bayesian filters in python," `https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python`, 9e3d2f6ed, 2014.

[18] 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.

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

[20] C. Zach, T. Pock, and H. Bischof, "A duality based approach for realtime tv-l 1 optical flow," in *Pattern Recognition: 29th DAGM Symposium*. Springer, 2007, pp. 214–223.

[21] J. Sánchez Pérez, E. Meinhardt-Llopis, and G. Facciolo, "TV-L1 Optical Flow Estimation," *Image Processing On Line*, vol. 3, pp. 137–150, 2013, `https://doi.org/10.5201/ipol.2013.26`.

[22] Z. Teed and J. Deng, "Raft: Recurrent all-pairs field transforms for optical flow," in *Computer Vision–ECCV 2020: 16th European Conference*. Springer, 2020, pp. 402–419.

[23] C. Dupre and R. Yuste, "Non-overlapping neural networks in hydra vulgaris," *Current Biology*, vol. 27, no. 8, pp. 1085–1097, 2017.

[24] N. Chenouard, I. Smal, F. De Chaumont, et al., "Objective comparison of particle tracking methods," *Nature methods*, vol. 11, no. 3, pp. 281–289, 2014.

[25] J. Huth, M. Buchholz, J. M. Kraus, et al., "Significantly improved precision of cell migration analysis in time-lapse video microscopy through use of a fully automated tracking system," *BMC cell biology*, vol. 11, pp. 1–12, 2010.

[26] J.-C. Olivo-Marin, "Extraction of spots in biological images using multiscale products," *Pattern Recognition*, vol. 35, no. 9, pp. 1989–1996, 2002.

[27] J. Schindelin, I. Arganda-Carreras, E. Frise, et al., "Fiji: an open-source platform for biological-image analysis," *Nature methods*, vol. 9, no. 7, pp. 676–682, 2012.

[28] J.-Y. Tinevez, N. Perry, J. Schindelin, et al., "Trackmate: An open and extensible platform for single-particle tracking," *Methods*, vol. 115, pp. 80–90, 2017.

[29] F. De Chaumont, S. Dallongeville, N. Chenouard, et al., "Icy: an open bioimage informatics platform for extended reproducible research," *Nature methods*, vol. 9, no. 7, pp. 690–696, 2012.

[30] J. Luiten, A. Osep, P. Dendorfer, et al., "Hota: A higher order metric for evaluating multi-object tracking," *International journal of computer vision*, vol. 129, pp. 548–578, 2021.

[31] A. Hanson, R. Reme, N. Telerman, et al., "Automatic monitoring of whole-body neural activity in behaving hydra," *bioRxiv*, pp. 2023–09, 2023.

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