Tracking Correction Method for Rapid and Random Protein Molecules Movement

Satoshi Kamiya[1], Keisuke Toida[1], Taka-aki Tsunoyama[2], and Kazuhiro Hotta[1]

[1] Meijo University, 1-501 Shiogamaguchi, Tempaku-ku, Nagoya 468-8502, Japan
[2] Okinawa Institute of Science and Technology Graduate University, 1919-1 Tancha, Onna-son, Kunigami-gun, Okinawa 904-0495, Japan
{180442042,190442098}@ccalumni.meijo-u.ac.jp kazuhotta@meijo-u.ac.jp

Abstract. In recent years, there has been a increasing demand for tracking protein molecules with the focus on immune system researches. However, machine learning-based single-particle tracking (SPT) faces the challenges in accuracy due to the rapid and random movement of molecules as well as detection errors. To address these issues, we use frame interpolation to pseudo-decrease the speed of movement and perform two-stage matching to achieve stable tracking. We also use an optimization algorithm that connects short tracks. This approach has achieved higher performance on the CD47 dataset and the PTC dataset than conventional baselines.

Keywords: single-particle tracking · multi-object tracking · optimization algorithm

1 Introduction

Single-Particle Tracking (SPT) is crucial for analyzing molecular trajectories by tracking multiple particles within microscope images. Traditional methods often detect molecular positions using techniques like Otsu's binarization [20], followed by tracking algorithms based on particle properties. However, these conventional methods suffer significantly from decreased accuracy due to variations in observation equipment and particle behavior, often necessitating manual corrections for analysis.

Fig. 1 illustrates the appearance of observed particles, which manifest as bright spots with intensities that fluctuate randomly over time. Additionally, due to the high density of protein molecules (100-500 molecules per video), occlusions where objects overlap and frequent tracking ID switching occur.

These detection errors often happen in a single frame, with fewer errors spanning multiple frames. Additionally, protein molecules move almost randomly, making their motion unpredictable and tracking challenging. In our experiments, we used a protein molecule called CD47, which moves approximately 7-15 pixels within one frame, presenting a considerable tracking challenge due to its rapid movement.

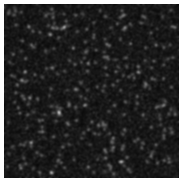


Fig. 1: Appearance of a protein molecule

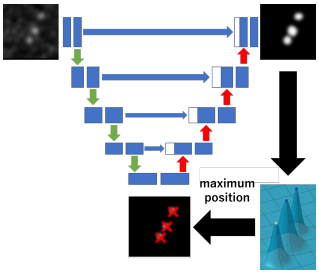


Fig. 2: Particle detection using UNet involves generating a probability map where the model is trained to predict the positions of object centers where the probability is highest. By identifying local maxima in the probability map, cells or molecules can be detected.

On the other hand, Machine Learning-based Multi-Object Tracking (MOT) has shown success in human and vehicle tracking. Machine learning-based tracking typically employs neural network architectures such as Convolutional Neural Networks (CNN) [15] and Transformers [24]. CNNs excel at extracting local information from input images using convolutions, enabling detailed position acquisition [12]. The prevailing method is "tracking by detection," where CNNs detect object positions, followed by algorithms for object tracking. Object detection involves enclosing features obtained from CNNs into various shapes of bounding boxes (bbox) and compressing them into a single feature vector using region of interest (RoI) pooling [10]. The confidence in the presence of an object within the enclosed area and precise position inference are then determined. This method enables the simultaneous detection of multiple objects and fast inference. Subsequently, common methods [3,8,26,28] for tracking involve techniques like Kalman filters [13] for motion prediction. However, existing tracking methods struggle with detecting small objects and tracking rapidly moving or irregularly moving objects, leading to decrease the accuracy, especially when dealing with small bright spots like protein molecules that move irregularly.

This paper proposes three methods to address the challenges of tracking protein molecules. The first method employs frame interpolation to detect molecules between frames, artificially reducing their movement speed to simplify tracking. The second method uses a two-stage matching process to correct single-frame detection errors. Finally, an optimization algorithm is employed to connect short-term tracks and refine the entire tracking process. The first two techniques enhance tracking accuracy at the local time scale, while the optimization algorithm allows for the correction of tracking errors at the global time scale.

We conducted experiments on CD47 protein molecule dataset and PTC dataset, comparing our approach against three conventional methods. Across all staining methods (GFP, TMR, SF650), our approach achieved higher accuracy compared to conventional methods. Furthermore, on the PTC dataset with

100 fluorescent spots, the IDF1 score improved from 75.4% to 92.7%. In the case of 500 fluorescent spots, the IDF1 score increased from 58.7% to 75.8%.

Contribution of this paper are as follows.

- In the detection using 3D UNet as described in Sec. 3.1, the training enhancements illustrated in Fig. 3 enable us to obtain detection results for non-existent frames. This method allows end-to-end detection and frame interpolation without additional annotations.
- We propose a two-stage matching based on frame differences in Sec. 3.2. This method primarily aims to address temporary detection errors.
- We propose a tracklet association algorithm in Sec. 3.3. Conventional methods only considered object splitting events and ignored fusion events. Our approach introduces fusion events by assuming that overlapping objects due to occlusion are in a pseudo-fusion state. Additionally, we propose an optimized process score suitable for protein movements.

2 Related Works

Recently, anchor-free object detection methods that do not use RoI pooling have been proposed. FCOS [23], YOLOX [9], and CenterNet [29] are representative anchor-free methods that achieve detection by predicting the size of bounding boxes and distinguishing between background and objects in all regions.

However, it is known that bbox-based object detection models have lower accuracy in detecting small objects, making them unsuitable for very small objects like protein molecules [1,27]. To address this, Nishimura et al. [19] successfully developed a Cell Detection model using UNet [21], which can obtain detailed position information. In this method, UNet is trained to output probability maps of objects, enabling anchor-free detection. The prediction flow is illustrated in Fig. 2. In this paper, we further improved detection accuracy by incorporating temporal information into the method, achieving enhanced tracking accuracy through temporal sequence interpolation.

In recent years, the research of Multi Object Tracking (MOT) has been actively pursued. Tracking-by-detection algorithms, which utilize detections and perform tracking using Kalman filters [13], have gained popularity. Well-known examples include SORT [3], DeepSORT [26], and StrongSORT [8]. Recently, ByteTrack [28], an improved version of SORT, was proposed as a method capable of high-speed and high-precision tracking. ByteTrack employs a two-stage matching approach based on detection scores to efficiently utilize the detected objects, achieving high accuracy in human tracking. However, ByteTrack relies on linear prediction using Kalman filters, making it unsuitable for tracking protein molecules with nonlinear motion.

Recently, end-to-end methods using Transformers [24] have gained attention in MOT, with various approaches like TrackFormer [18] and TransTrack [22] being proposed. Transformer-based methods utilize neural networks to perform tracking using surrounding object information and positional data, allowing for

capturing nonlinear motion. However, these methods often suffer from the compression of spatial features in the feature maps, making it difficult to obtain precise positional information and effectively track small objects.

ByteTrack and TransTrack focus on real-time MOT, prioritizing inference speed and the ability to process data in real-time without correcting past tracking results. Recently, tracking has been increasingly employed for data analysis in research, leading to a growing demand for tracking methods that prioritize the accuracy over inference speed. SUSHI [5] achieved high accuracy by performing both short-term and long-term tracking using past tracking data. Additionally, Bise et al. [4] proposed a method to generate improved tracks by connecting track fragments after completing tracking for the entire video. These methods are known for their robustness against occlusions and tracking interruptions caused by detection errors.

Cell tracking methods, such as those employing graph theory by Ben-Haim [2] and the Moving Point Model (MPM) [11], have been proposed. However, cell tracking differs significantly from tracking protein molecules. Cells have distinct shapes, sizes, and intensities, making individual identification possible. Conversely, protein molecules are observed as bright spots with limited individual differences, making their tracking challenging. Moreover, protein molecules exhibit irregular motion, making movement prediction difficult and causing decreased accuracy in methods like MPM that rely on predictable trajectories.

In this paper, we propose a tracking method with three key components: particle detection using frame interpolation, a two-stage tracking process between real frames and interpolated frames, and a global tracking algorithm that connects short-term tracks. The first two components enable robust tracking at local timescales, allowing the method to accommodate rapid protein movements. Furthermore, the third component corrects tracking errors and produces reliable tracking outputs through global optimization across the entire video sequence.

3 Proposed Method

Fig. 3 shows the overview of our method. The method consists of three main components: detection using frame interpolation described in Sec. 3.1, a two-stage matching method detailed in Sec. 3.2, and an optimization technique to connect short-term tracks and correct the overall tracking results. The global tracking algorithm is described in Sec. 3.3.

3.1 Detection Using Frame Interpolation

The objective of the proposed method is to reduce the movement distance between frames using frame interpolation, thereby simplifying the tracking process. Instead of a standard 2D UNet, the model employs a 3D UNet [7]. Utilizing a 3D UNet allows for capturing temporal information in video sequences, thus improving detection accuracy. The detection method is based on the approach

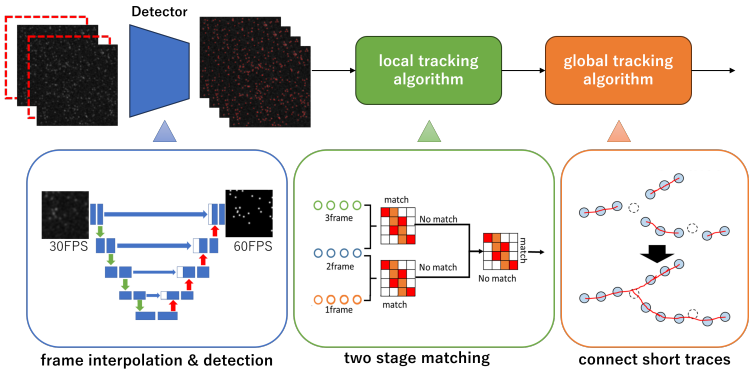


Fig. 3: Overview of our method: Firstly, we generate detection maps for 2N frames using a 3D UNet [7], applied to N frames of video data for molecular detection. Next, we employ a two-stage matching process for each frame using the expanded detections. Finally, we apply global optimization to rectify short-term tracking errors and derive the tracking results.

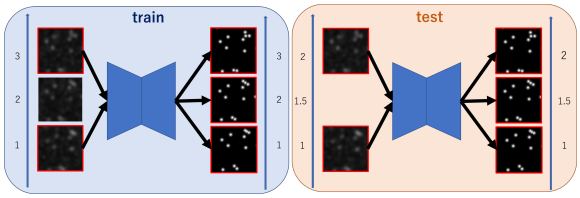


Fig. 4: Training and inference with 3D UNet: The training data consists of video sequences captured at 30 frames per second (fps). During training, video sequences at 15 fps is fed into the model and the model learns to output detection results at 30 fps. During inference, a video captured at 30 fps is fed into the model, which then outputs detection results at 60 fps for tracking purposes.

proposed by Nishimura et al. [19], with adjustments made to facilitate the detection of maximal positions. In conventional methods, there was a problem of detecting all maximal positions when multiple local maxima existed for the same molecule, leading to false detections. However, the proposed method determines non-overlapping regions when detecting maximal positions, and it controls the detection process so that only one object is detected in each region.

Fig. 4 illustrates the overview of training and inference with the 3D UNet. A video of N frames is fed into the 3D UNet, which is trained to output particle detection results for 2N frames. To interpolate the video from N frames to 2N frames, convolutional processing is used for frame interpolation. The 3D UNet is trained to output high confidence scores at the central positions of molecules. During inference, the positions of maximal confidence correspond to the molecular positions. Since the video contains molecules with various velocities, the trained model becomes robust to different molecular speeds. Consequently, detec-

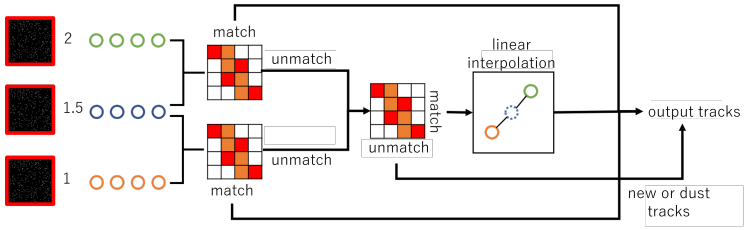


Fig. 5: Two-stage matching algorithm

tion at different frame rates during inference does not pose a problem. Therefore, training can use video data with similar frame rates to those used in inference, eliminating the need to create separate data for training purposes.

3.2 Two-stage Matching Based on Frame Difference

The purpose of the two-stage matching algorithm is to mitigate tracking interruptions caused by detection errors. Inspired by ByteTrack [28], our two-stage matching method is designed to ensure stability and handle detection errors effectively. While ByteTrack divides tracking into two stages based on detection scores, our proposed method divides the matching process into two stages based on frame differences.

The tracking algorithm of the proposed method is illustrated in Fig. 5. It demonstrates the process of detecting and tracking objects for frames 1, 1.5, and 2 using the approach described in Sec. 3.1. Matching is performed using the Hungarian algorithm [16] for one-to-one matching, seeking pairs that minimize the distance between objects.

In the initial matching stage, matching is performed between frames 1 and 1.5, and between frames 1.5 and 2. For unmatched objects, tracking is conducted between frames 1 and 2. This approach can utilize pseudo-temporal information. Additionally, it is known that object motion can be approximated as linear over small time intervals, allowing for the consideration of simple linear movements even for randomly moving objects like protein molecules [3]. Objects that cannot be tracked in the 0.5 frame difference matching are likely due to detection errors. Therefore, 1 frame difference matching is employed to prevent tracking interruptions and generate long-term tracks. For tracks matched with 1 frame difference, where there are no detections in between, spline interpolation [17] is utilized to generate intermediate detections and correct the tracking.

3.3 Global Tracking Algorithm

The global tracking algorithm aims to consider all short-term tracking results and optimally connect them to perform long-term tracking. For overall correction, our method is based on the approach proposed in [4], as illustrated in 6. In this method, short-term tracks have start and end points, and they are either

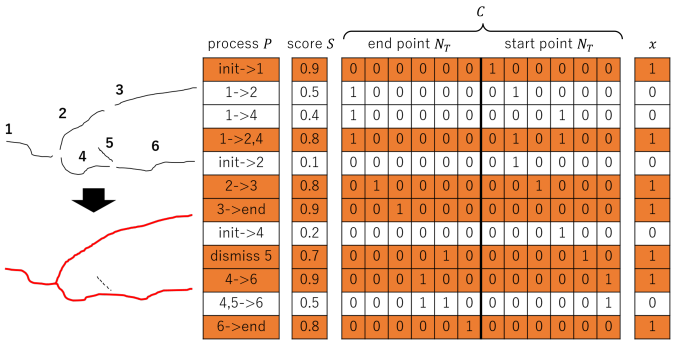


Fig. 6: Global tracking algorithm

connected with other tracks or left disconnected to generate longer tracks. This approach converts these connection decisions into a linear optimization problem to produce the optimal tracking results.

The conventional method [4] only handles cell division and does not address fusion events. In the case of small bright spots like protein molecules, they are observed to pseudo-fuse due to occlusion. Thus, the conventional method does not account for these pseudo-fusions caused by occlusion. Additionally, in the conventional method, optimization is performed based on the objects detected initially in the sequence. Consequently, if the initial detection fails, the entire tracking process fails, leading to instability. Therefore, our proposed method addresses fusion events and introduces an optimization technique that evenly considers all tracks, including fusion events.

In the existing method [4], a tree structure is created with the detections from the initial frame as the starting points. Each track is then optimized within its respective tree structure to ensure no overlaps occur between tracks. In contrast, our proposed method does not create a tree structure; instead, it optimizes all tracks uniformly.

The optimization method is defined by the process score vector $p \in \mathbb{R}^{N_p}$, the process matrix $C \in 0, 1^{N_p \times 2N_T}$, and the selection vector $x \in 0, 1^{N_p}$ as

$$\max_x \; p^T x \quad s.t.\; C^T x = [1]^{2N_T}, \tag{1}$$

where N_p and N_T respectively represent the number of processes and the total number of tracks. The selection vector x is defined such that its elements are 0 when a process is not selected and 1 when it is selected. There are six types of processes for each process p: track appearance, track continuation, track splitting, track merging, track disappearance, and track discard, where the last one indicates unused tracks.

The process matrix and scoring method are illustrated in Fig. 7. The left side depicts a scenario with four tracks, and the resulting process matrix from applying six processes is shown on the left, while the method for determining process scores is presented on the right. For example, to represent the process

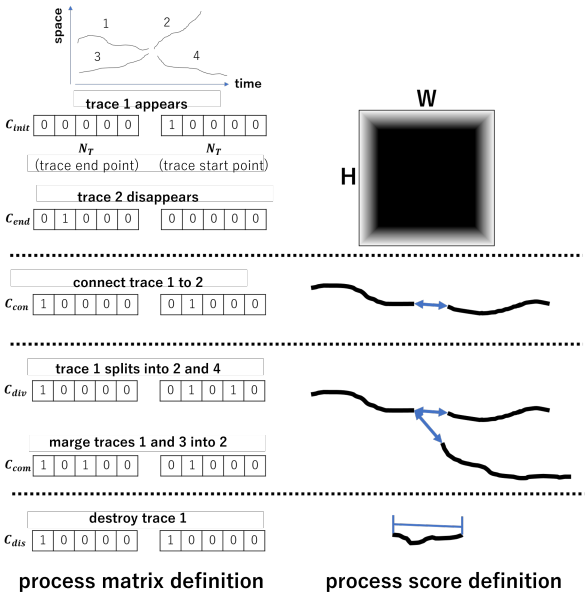


Fig. 7: Process matrix and score definition

of track 1 appearing in the process matrix, the left side of the matrix contains the endpoint numbers of the tracks, while the right side represents the starting point numbers. The track appearance process, where a track's starting point is not connected to any endpoint, results in the first column of the process matrix being set to 1 for track 1, as its endpoint is not connected. Similarly, the process matrix for track 2's disappearance has its endpoint unconnected, leading to the second column being set to 1 in the first half, and all zeros in the second half. This representation allows each process to be expressed as a column in the matrix.

Additionally, appearances and disappearances occur as molecules move off-screen, with the likelihood of these events increasing as molecules approach the screen's edge. Therefore, given the positions of a track's start and end points as x and y, the appearance and disappearance scores p_{init} and p_{end} are defined as follows:

$$d_{edge}(x, y) = min(x, y, W - x, H - y) \tag{2}$$

$$p_{init}, p_{end} = max(exp(-\frac{d_{edge}(x, y)}{\tau_{init}}), 0.05) \tag{3}$$

where τ_{init} represents the temperature parameter, and H and W denote the dimensions of the input image. The function $d_{edge}(\cdot)$ indicates the shortest distance from the given coordinate to the screen edge. This approach allows for tracking while considering objects likely to appear or disappear at the screen edge. However, in the case of protein molecules, while they often appear or disappear from the screen's edge, sometimes the activation of a luminous point may

cause appearances or disappearances from the center of the screen. Therefore, in the equations above, high scores are assigned to the screen edges, while smaller values are given to the center to account for these cases.

When connecting short-term tracks, linking the ending points of one track to the starting points of another is necessary. For instance, when connecting track 1 and track 2 as shown in Fig. 7, the first column is set to 1 for track 1's endpoint, and the second column is set to 1 for track 2's starting point. Additionally, the connection score is determined based on the distance between the ending point of one track and the starting point of the next, such that smaller distances result in higher connection scores p_{con}.

$$p_{con} = exp(-\frac{d(x_i, y_i, x_j, y_j)}{d_{max}}) \tag{4}$$

where $d(\cdot)$ represents the Euclidean distance between two coordinates, d_{max} is the maximum displacement of molecules obtained from the training data, and x_i, y_i denote the xy coordinates of the i-th track's endpoint, while x_j, y_j represent the starting point of the j-th track. Evaluation of the distance between tracks helps prevent unnatural connections between them.

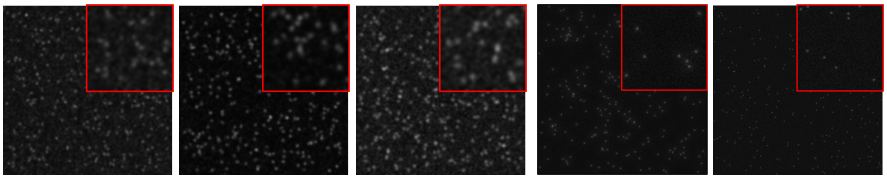
Moreover, there are two starting points to link in track splitting. For example, if track 1 splits into tracks 2 and 4 as shown in Fig. 7, the first column is set to 1 for track 1's endpoint, and the second and fourth columns are set to 1 for the starting points of tracks 2 and 4, respectively. Similarly in the process of merging, since there are two endpoints to connect to one starting point, the process matrix resembles the one shown in Fig. 7. The scores for splitting and merging, p_{div} and p_{com} are evaluated based on the distances between the tracks, similar to the connection score.

$$p_{div}, p_{com} = exp(-\frac{d(x_i, y_i, x_j, y_j)}{d_{max}}) + exp(-\frac{d(x_i, y_i, x_k, y_k)}{d_{max}}) \tag{5}$$

In the case of track disposal, we require a process matrix indicating "unused". However, if all elements in the process matrix are set to 0, it would violate the constraint $C^T x = [1]^{2N_T}$, indicating that no selections are made. Therefore, when track 1 is discarded for instance, a process matrix is defined to link track 1's starting point and endpoint as shown in Fig. 7. The constraint given by Eq. (1) ensures that once a starting or ending point is used it cannot be reused. Thus, by defining a process matrix that makes it impossible to use both the starting and ending points of a track simultaneously, track disposal is effectively represented.

Additionally, the disposal score p_{dis} allows for the evaluation of tracks based on the length of consecutive frames, helping to determine whether a track is noise-derived or not [25]. To achieve these objectives, p_{dis} is designed to inversely correlate with the track's consecutive frame length. Specifically, shorter tracks are assigned higher disposal probabilities, as they are more likely to represent noise or detection errors. This relationship is mathematically expressed as:

$$p_{dis} = exp(-\frac{l}{\tau_{dis}}) \tag{6}$$



(a) GFP **(b)** TMR **(c)** SF650

Fig. 8: CD47 dataset

(a) vesicle **(b)** receptor

Fig. 9: PTC dataset

where τ_{dis} is the temperature parameter and l denotes the number of consecutive frames for a track.

The process matrix and scores are determined as described, with the results adapted for each individual track depicted in Fig. 6. This figure illustrates a subset of processes applied to a given set of tracks, visible on the left side. The process matrix and scores are calculated using the previously outlined methods, and the selection vector x optimizes process selection to maximize the overall score. A constraint is imposed on the selection vector x, ensuring that the sum of selected processes' column-wise totals in the process matrix equals 1. In Fig. 6, orange columns summing to 1 represent this constraint, indicating that each starting or ending point can be used only once. Consequently, the number of 1s in the rows of the process matrix becomes a critical factor. Processes consuming multiple starting and ending points, such as splits and mergers, inherently receive lower scores relative to the number of points they consume. To compensate for this, p_{div} and p_{com} in Eq. (5) are calibrated to yield higher scores compared to other processes.

4 Experiments

4.1 Setting

Datasets. In our experiments, we utilize simulated videos of CD47 protein molecule movement and simulated videos of vesicles and receptors from the Particle Tracking Challenge (PTC2012) [6]. This approach is necessitated by the difficulty in assigning ground truth to real protein molecule videos. Fig. 8 illustrates the CD47 dataset. This dataset allow for the adjustment of particle density and staining methods, including GFP (Fig. 8a), TMR (Fig. 8b), and SF650 (Fig. 8c). CD47 exhibits alternating periods of random movement and attachment to other protein molecules, resulting in temporal variations in the intensity of fluorescent spots. Additionally, protein molecules occasionally become completely invisible for approximately one frame, presenting significant challenges for tracking algorithms.

Fig. 9 provides the overview of the PTC dataset. Vesicles (Fig. 9a) exhibit random movement similar to protein molecules but with reduced mobility, and their fluorescent spots appear smaller than those in the CD47 dataset. Receptors (Fig. 9b) move linearly, but they are even smaller than vesicles, presenting greater

challenges for detection. The difficulty of tracking varies depending on the density of fluorescent spots in the image, with higher densities generally increasing the complexity of the tracking task.

For both CD47 and PTC2012 videos, the Signal-to-Noise Ratio (SNR) can be adjusted, representing the ratio of fluorescent spot intensity to noise intensity. In PTC2012, the SNR is fixed at 7 for all experiments, in accordance with the original PTC2012 paper. However, for the CD47 dataset, the SNR varies depending on the staining method employed. We utilize three staining methods: GFP, TMR, and SF650. GFP staining (Fig. 8a) results in a low SNR, making fluorescent spot distinction challenging, while SF650 (Fig. 8c) staining facilitates easier spot identification. The SNR of TMR (Fig. 8b) falls approximately midway between GFP and SF650. We train and evaluate models using videos created with each of these staining methods. For the ablation study, we utilize the standard CD47 dataset with GFP staining, which represents the most challenging dataset.

Implementation details. Our experiments involve tracking using images containing either 100 or 300 spots. Each video consists of 100 frames with an image size of 512x512 pixels. We prepare five sets of videos, each containing either 100 or 300 spots. Three sets of 300-spot videos are used as training data. One set of 100-spot videos serves as validation data, while the remaining sets (both 100 and 300-spot videos) are used for evaluation. We employ 5-fold cross-validation to compare the accuracy of each method. Training is conducted for 200 epochs using Cosine annealing and Adam optimization. For inference, τ_{init} and τ_{dis} are set to 3 and 1, respectively. Our comparative analysis includes: TrackFormer [18], a Transformer-based tracking model, MPM [11], a machine learning model for Cell Tracking, and PTGT [14], which enables long-term tracking using Transformer and tracking algorithms.

4.2 Comparisons with Other Association Methods

Tab. 1 presents the tracking accuracy of each method for 100 and 300 tracked objects. IDF1, IDP, and IDR represent ID F1 score, ID precision, and ID recall, respectively. "ALL" denotes the average accuracy across all staining methods. The proposed method achieved superior accuracy compared to other methods across all staining methods. Tracking accuracy significantly improved as the staining methods became more favorable for detection. This improvement is attributed to easier detection, which facilitates better frame-to-frame correspondence, thus reducing pseudo motion with frame interpolation. Tracking becomes notably challenging with 300 objects, resulting in a higher number of ID switches. However, the proposed method exhibited the lowest ID switch rate and achieved high overall accuracy. PTGT also shows a tendency towards lower ID switches, indicating the importance of mechanisms for correcting tracking interruptions in protein tracking scenarios. Tab. 1 also demonstrates the detection accuracy for 300 objects. F1, Pr, and Re represent detection F1 score, precision, and

Table 1: Comparison of tracking and detection accuracy on CD47 dataset.

Num of objects		100				300						
Staning	Method	IDF1	IDP	IDR	IDsw	IDF1	IDP	IDR	IDsw	F1	Pr	Re
GFP	TrackFormer	22.50%	19.29%	27.20%	1095	12.76%	10.00%	17.63%	6240	66.73%	52.29%	**92.20%**
	MPM	43.35%	44.47%	42.28%	557	29.64%	33.34%	26.68%	2718	86.49%	97.20%	77.90%
	PTGT	65.46%	51.22%	90.71%	**71**	46.14%	47.19%	45.14%	462	**92.90%**	98.43%	87.97%
	ours	**69.55%**	**57.36%**	**92.84%**	78	**50.22%**	**55.35%**	**45.96%**	**415**	92.27%	**99.73%**	85.84%
TMR	TrackFormer	25.75%	23.97%	28.01%	911	12.69%	9.84%	17.87%	6298	65.78%	51.00%	**92.66%**
	MPM	59.88%	**62.18%**	57.75%	336	33.07%	37.36%	29.67%	2554	87.62%	98.97%	78.61%
	PTGT	66.53%	52.49%	91.88%	**72**	46.50%	48.72%	44.48%	466	**93.22%**	98.59%	88.41%
	ours	**70.85%**	57.66%	**93.45%**	81	**56.31%**	**59.32%**	**55.48%**	**389**	92.67%	**99.59%**	86.64%
SF650	TrackFormer	12.42%	8.97%	20.16%	1916	10.08%	7.40%	15.77%	7036	60.51%	44.44%	**94.76%**
	MPM	55.24%	**57.27%**	53.36%	372	31.19%	35.49%	27.82%	2582	86.76%	98.70%	77.41%
	PTGT	64.19%	49.55%	91.74%	**65**	46.17%	48.33%	44.20%	432	92.81%	98.59%	87.67%
	ours	**67.82%**	56.12%	**94.44%**	72	**63.19%**	**66.33%**	**60.34%**	**378**	**94.97%**	**99.69%**	90.68%
ALL	TrackFormer	20.82%	20.90%	23.26%	1054	11.84%	9.08%	17.09%	6525	64.34%	49.24%	**93.21%**
	MPM	42.56%	43.48%	41.95%	782	31.30%	35.40%	28.06%	2618	86.96%	98.29%	77.97%
	PTGT	61.24%	47.01%	89.09%	**69**	46.27%	48.08%	44.61%	453	92.98%	98.54%	88.02%
	ours	**69.41%**	**57.05%**	**93.58%**	77	**56.57%**	**60.33%**	**53.93%**	**394**	**93.30%**	**99.67%**	87.72%

Table 2: Comparison of tracking accuracy on PTC dataset.

Num of objects		100				500			
Molecule	Method	IDF1	IDP	IDR	IDsw	IDF1	IDP	IDR	IDsw
RECEPTOR	TrackFormer	40.14%	34.90%	47.31%	494	42.26%	37.44%	48.53%	2336
	MPM	67.13%	69.04%	65.31%	442	60.64%	66.29%	55.87%	1571
	PTGT	71.92%	68.46%	75.98%	**33**	57.12%	44.24%	**80.58%**	**309**
	ours	**93.84%**	**96.50%**	**91.33%**	38	**76.83%**	**81.21%**	72.99%	862
VESICLE	TrackFormer	37.80%	29.42%	52.88%	1207	35.12%	28.60%	45.49%	4655
	MPM	65.64%	68.15%	63.32%	517	55.91%	62.51%	50.57%	2291
	PTGT	78.87%	81.29%	76.59%	**29**	60.19%	55.41%	65.89%	**195**
	ours	**91.57%**	**94.37%**	**88.94%**	70	**74.68%**	**81.36%**	**69.01%**	1307
ALL	TrackFormer	38.97%	32.16%	50.09%	851	38.69%	33.02%	47.01%	3496
	MPM	66.39%	68.60%	64.32%	480	58.27%	64.40%	53.22%	1931
	PTGT	75.39%	74.88%	76.28%	**31**	58.65%	49.82%	**73.24%**	**252**
	ours	**92.71%**	**95.43%**	**90.13%**	54	**75.75%**	**81.28%**	71.00%	1085

recall, respectively. MPM, PTGT, and the proposed method, employing UNet models for detection, exhibit remarkably high detection accuracy compared to TrackFormer, which does not use UNet. Despite both PTGT and the proposed method utilizing 3D UNet, the proposed method outperforms PTGT in detection accuracy. This is attributed to the consideration of finer temporal information, usually overlooked, thanks to frame interpolation. While the difference in overall detection accuracy is less than 1%, the 10.30% improvement in overall tracking accuracy indicates the effectiveness of two-stage matching methods and global tracking algorithms.

Next, we present the accuracy on the PTC dataset. Tab. 2 shows the tracking accuracy of each method in videos with 100 and 500 fluorescent spots. "RECEPTOR" and "VESICLE" represent receptors and vesicles, respectively. Tab. 2 indicates that the proposed method achieved exceptionally high accuracy and can track almost all particles effectively. In the PTC dataset, particles frequently appear and disappear, leading to numerous short-term tracking instances. Consequently, methods like MPM, which struggle with predicting movements immediately after appearance, and PTGT which sometimes connect tracking to other particles after disappearance, experience decreased accuracy. The improved ac-

Table 3: Ablation study by adding proposed methods such as Detection using Frame Interpolation(DFI), Two-Stage Matching(TSM) and Global Tracking Algorithm(GTA). We use the CD47 dataset with 300 fluorescent spots stained with GFP.

	DFI	TSM	GTA	IDF1	IDPr	IDRe
(baseline)				43.76%	49.63%	40.26%
	✓			45.12%	50.47%	42.66%
	✓	✓		47.46%	52.31%	43.43%
(ours)	✓	✓	✓	**50.22%**	**55.35%**	**45.96%**

Table 4: The decrease in detection accuracy due to frame interpolation.

frame	F1	Pr	Re
all	92.27%	99.73%	85.84%
real	93.01%	99.74%	87.13%
interpolated	91.21%	99.48%	84.21%

Table 5: The influence in tracking accuracy due to the decreased detection accuracy.

F1	IDF1	IDP	IDR	ID_switch
92.27%	50.22%	55.35%	45.96%	415
91.27%	50.18%	55.32%	45.96%	415
90.28%	50.21%	55.41%	45.76%	412
89.26%	49.74%	54.77%	43.86%	432

curacy of the proposed method is attributed to consider the overall tracking results in order to determine optimal connection methods. With 500 fluorescent spots, conventional methods like MPM exhibit poor accuracy in vesicles due to movement prediction, whereas they perform better with receptors. On the other hand, PTGT excels in vesicles due to its strength in handling random movements but lags behind MPM in accuracy with receptors. The proposed method demonstrates high accuracy in both scenarios, indicating resilience to the influence of object movement types. Objects can be approximated with linear movements when considering short time intervals [3]. Therefore, the detection method with frame interpolation can capture objects as simple linear movements, enabling tracking regardless of the object's movement pattern.

4.3 Ablation Studies

This section aims to validate the effectiveness and validity of our method. The GFP from the CD47 dataset is used for validation purposes because it is the most challenging dataset. Tab. 3 shows the changes in accuracy due to the presence or absence of the proposed method. The baseline employs a 3D UNet for detection and utilizes Hungarian matching for one-to-one tracking. When tracking is performed using the frame interpolation detection method, there is an improvement in accuracy by approximately 1.36%. Further enhancements are achieved by incorporating a two-stage tracking method, resulting in a 2.34% increase in accuracy. Additionally, an optimization technique for connecting short-term tracks contributes the most to the accuracy, achieving a 2.76% improvement. This indicates its significant contribution among the proposed approaches.

Although this paper simplifies tracking by detecting objects in frames that would not naturally exist through frame interpolation, validation examines whether the detection accuracy of non-existent frames through frame interpolation has decreased. Validation method obtains detection results of 30 fps from a video of

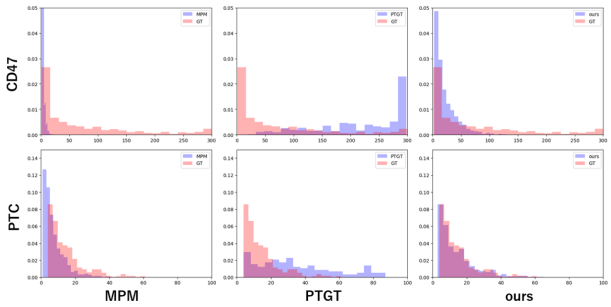


Fig. 10: Histogram of the tracking lengths for both predicted and ground truth data. The horizontal axis represents the length of the tracks, while the vertical axis indicates the number of tracks. The blue histogram represents the predicted results while the red one represents the ground truth data. The prediction results by the proposed method the best match to the distribution of the ground truth data.

15 fps using the model. Subsequently, the detection accuracy of input frames is compared with that of the interpolated parts to assess the validity of the method. Tab. 4 shows the detection accuracy by frame interpolation. The overall accuracy represents the combined accuracy of the input and interpolated parts. The difference in accuracy between the input and interpolated parts is approximately 1.8%, raising concerns about the validity of this accuracy reduction.

To assess the validity of the decrease in detection accuracy, Tab. 5 shows the change in tracking accuracy when detection accuracy is decreased. The detection accuracy is reduced by 1% by randomly removing overall detection results and adding noise. In Tab. 5, when detection accuracy drops by about 3%, the tracking results significantly deteriorate while there is little change in accuracy before that. This is attributed to the correction of areas not detected by the two-stage tracking method or the global tracking method, which helps to mitigate the decrease in accuracy. Therefore, the decrease in detection accuracy due to frame interpolation has minimal adverse effects on tracking results, indicating that detection via frame interpolation is a valid technique.

Fig. 10 show the histograms of the tracking lengths for both predicted and ground truth data. The prediction distribution by the proposed method closely matches the ground truth.

5 Conclusion

We present a tracking method by integrating the detection by frame interpolation, two-stage matching algorithm, and global tracking algorithm. The effectiveness is confirmed by the experiments on CD47 and PTC dataset.

Acknowledgements

This work was partially supported by SCAT research grant.

References

1. Bai, Y., Zhang, Y., Ding, M., Ghanem, B.: Sod-mtgan: Small object detection via multi-task generative adversarial network. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 206–221 (2018)
2. Ben-Haim, T., Raviv, T.R.: Graph neural network for cell tracking in microscopy videos. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXI. pp. 610–626. Springer (2022)
3. Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B.: Simple online and realtime tracking. In: 2016 IEEE international conference on image processing (ICIP). pp. 3464–3468. IEEE (2016)
4. Bise, R., Yin, Z., Kanade, T.: Reliable cell tracking by global data association. In: 2011 IEEE international symposium on biomedical imaging: From nano to macro. pp. 1004–1010. IEEE (2011)
5. Cetintas, O., Brasó, G., Leal-Taixé, L.: Unifying short and long-term tracking with graph hierarchies. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22877–22887 (2023)
6. Chenouard, N., Smal, I., De Chaumont, F., Maška, M., Sbalzarini, I.F., Gong, Y., Cardinale, J., Carthel, C., Coraluppi, S., Winter, M., et al.: Objective comparison of particle tracking methods. Nature methods **11**(3), 281–289 (2014)
7. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3d u-net: learning dense volumetric segmentation from sparse annotation. In: International conference on medical image computing and computer-assisted intervention. pp. 424–432. Springer (2016)
8. Du, Y., Zhao, Z., Song, Y., Zhao, Y., Su, F., Gong, T., Meng, H.: Strongsort: Make deepsort great again. IEEE Transactions on Multimedia (2023)
9. Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J.: Yolox: Exceeding yolo series in 2021. arXiv preprint arXiv:2107.08430 (2021)
10. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 1440–1448 (2015)
11. Hayashida, J., Nishimura, K., Bise, R.: Mpm: Joint representation of motion and position map for cell tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3823–3832 (2020)
12. Islam, M.A., Jia, S., Bruce, N.D.: How much position information do convolutional neural networks encode? arXiv preprint arXiv:2001.08248 (2020)
13. Kalman, R.E.: A new approach to linear filtering and prediction problems (1960)
14. Kamiya, S., Hotta, K., Tsunoyama, T., Kusumi, A.: Single-particle tracking by graph transformer. In: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1–5. IEEE (2023)
15. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems **25** (2012)
16. Kuhn, H.W.: The hungarian method for the assignment problem. Naval research logistics quarterly **2**(1-2), 83–97 (1955)
17. McKinley, S., Levine, M.: Cubic spline interpolation. College of the Redwoods **45**(1), 1049–1060 (1998)
18. Meinhardt, T., Kirillov, A., Leal-Taixe, L., Feichtenhofer, C.: Trackformer: Multi-object tracking with transformers. arXiv preprint arXiv:2101.02702 (2021)

19. Nishimura, K., Ker, D.F.E., Bise, R.: Weakly supervised cell instance segmentation by propagating from detection response. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 649–657. Springer (2019)
20. Otsu, N.: A threshold selection method from gray-level histograms. vol. 9, pp. 62–66 (1979). https://doi.org/10.1109/TSMC.1979.4310076
21. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
22. Sun, P., Cao, J., Jiang, Y., Zhang, R., Xie, E., Yuan, Z., Wang, C., Luo, P.: Transtrack: Multiple object tracking with transformer. arXiv preprint arXiv:2012.15460 (2020)
23. Tian, Z., Shen, C., Chen, H., He, T.: Fcos: Fully convolutional one-stage object detection. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9627–9636 (2019)
24. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
25. Wen, L., Li, W., Yan, J., Lei, Z., Yi, D., Li, S.Z.: Multiple target tracking based on undirected hierarchical relation hypergraph. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1282–1289 (2014)
26. Wojke, N., Bewley, A., Paulus, D.: Simple online and realtime tracking with a deep association metric. In: 2017 IEEE international conference on image processing (ICIP). pp. 3645–3649. IEEE (2017)
27. Yang, F., Choi, W., Lin, Y.: Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2129–2137 (2016)
28. Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., Wang, X.: Bytetrack: Multi-object tracking by associating every detection box. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII. pp. 1–21. Springer (2022)
29. Zhou, X., Wang, D., Krähenbühl, P.: Objects as points. arXiv preprint arXiv:1904.07850 (2019)