DanioSense: Automated High-Throughput Quantification of Zebrafish Larvae Group Movement

Chunxiang Wang, Mingsi Tong, *Member, IEEE*, Liqun Zhao, Xinghu Yu, Songlin Zhuang, *Member, IEEE*, and Huijun Gao, *Fellow, IEEE*

***Abstract*— The capability to obtain detailed motility information of model organisms is fundamental to reveal their functional and social behavior characteristics. Zebrafish is a powerful vertebrate model organism. Despite recent success in the automatic quantification of adult zebrafish movement, it remains a laborious task for group zebrafish larval tracking due to their similar appearance, frequent occlusions, and highly discontinuous kinematics. This article presents DanioSense (DS), an automatic tracker for group larval zebrafish, to overcome these tracking challenges. The integration of a light convolutional neural network and a centerline extraction algorithm enables the tracker to localize individuals even in occlusion cases where objects' identities are prone to switch. With reliable detections, an adaptive Kalman filter is designed to optimally estimate locomotive parameters, which is also used for object reidentification accomplished by a two-stage data association protocol. Experimental results demonstrated a tracking accuracy of over 97%, median errors of 102 μm, and 8.8° for the position and orientation measurement, and a processing speed of over 30 frames/s with a normal computer configuration. DS provides detailed quantitative data for a large-scale larvae group in nearly real time, highly boosting the efficiency of characterizing individual phenotypes and analyzing social interactions.**

***Note to Practitioners*—This article aimed to tackle the problem of automated tracking groups of zebrafish larvae, an ideal vertebrate model organism for large-scale chemical and genetic screens. The task of group tracking is to record each individual's movement and calculate their position, velocity, direction, and other parameters for further analysis, where the correct identity of each individual must be maintained. Existing algorithms either switch larvae' identities easily or are unable to achieve online tracking due to the limitations of their methods to address individuals' intersections. DanioSense (DS) adopts a convolutional neural network to identify larval heads whenever they intersect and uses an adaptive Kalman filter to calculate the movement parameters optimally. Besides, a range of visualization options is designed to bring insight into underlying patterns through massive amounts of data. Theoretically, this algorithm's approach to solving intersections and calculating movement statistics can also apply to other fish-like animals. Its visualization options are applicable to other tracking systems. The key advantage of Daniosense over existing trackers is the capability to track each larva within a group and output detailed quantitative data in nearly real time. The tracking performance of DS is based on the quality of image segmentation and the success rate of classifying samples. Many state-of-the-art image segmentation and classification neural networks can be adopted to extend this system's applications to more complex environments but at a higher computation and time cost, which is a tradeoff between efficiency and capability. Some applications require a higher video sampling rate, so the system's processing speed needs to be further improved with better hardware and software framework optimization. The next steps include improving the processing efficiency, providing more tracking modules and visualization options, and extending its application fields.**

***Index Terms*— Automated measurement, multiple-object tracking, video analytics, zebrafish larvae.**

I. Introduction

VIDEO tracking has become a standard procedure for studying model organisms' functional characteristics, during which the trajectory of each animal and their corresponding movement information are extracted for a range of biomedical studies, such as genetics [1], drug discovery [2], toxicology [3], and behavioral science [4]. Manual tracking via frame-by-frame labeling is time-consuming, subjective, and tedious, whose results are often not consistent due to the fatigue and inexperience of operators. Therefore, an automatic tracking system is indispensable, which achieves a remarkedly higher research output and provides more insightful and detailed statistics.

Zebrafish (*Danio rerio*) larva stands out as a popular vertebrate model organism for large-scale chemical and genetic screens due to its high similarity of gene and cardiovascular system to human, optical transparency, easiness of acquisition, and rapid developmental process [5], [6]. Compared with other experimental organisms of zebrafish, larval motility, along with the transparent morphology, implies more phenotypic

Manuscript received August 26, 2020; revised November 1, 2020 and December 7, 2020; accepted December 21, 2020. This article was recommended for publication by Associate Editor L. Zhang and Editor C. Seatzu upon evaluation of the reviewers' comments. This work was supported in part by the National Natural Science Foundation of China under Grant 61933008. *(Corresponding author: Huijun Gao.)*

Chunxiang Wang, Mingsi Tong, Liqun Zhao, and Huijun Gao are with the Research Institute of Intelligent Control and Systems, Harbin Institute of Technology, Harbin 150001, China (e-mail: hjgao@hit.edu.cn).

Xinghu Yu is with Ningbo Institute of Intelligent Equipment Technology Corporation, Ningbo 315000, China (e-mail: yuxinghu1012@126.com).

Songlin Zhuang is with the Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, ON M5S 3G8, Canada (e-mail: songlinzhuang@hit.edu.cn).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TASE.2021.3050408.

Digital Object Identifier 10.1109/TASE.2021.3050408

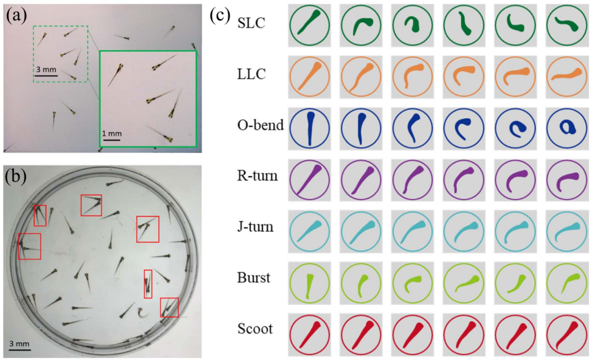

Fig. 1. (a) Small size and similar, transparent appearance. (b) Larval occlusion, enclosed by red rectangles, a critical problem for tracking. (c) Various swim modes, which lead to the poor performance of tracking algorithms based on continuous motion models and body geometry.

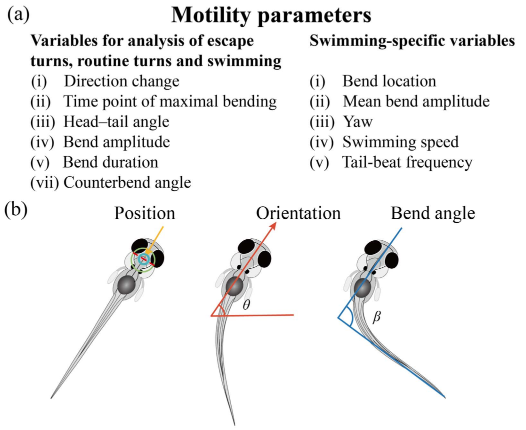

Fig. 2. (a) Motility parameters for further analysis. (b) Primary variables to track, position, orientation, and bend angle. Quantitative motility parameters are derived from these primary variables.

information than cells and embryos. Unlike adult zebrafish, larval relative transparency and immature nervous system allow detailed *in vivo* studies of gene regulation and function and neuropharmacological studies that cannot be performed in the adult period [7]. Researchers generally transfer foreign materials into zebrafish embryos [8]–[10] or specific organs of zebrafish larvae [11]–[15] after which larval locomotive behaviors are recorded for further analysis [16]. These investigations cover the fields of the analysis of movement characteristics [17], social behavior [18], and the assessment of phenotypes resulting from gene knockdown approaches [19], genetic mutations [20], drugs [21], [22], and so on.

Many single- and multiple-object tracking systems have been developed for adult zebrafish, reporting outstanding tracking performance [23]–[25]. However, their performance dramatically drops when applied to a larvae group, where individuals' identities switch easily, and thus, the statistical data are not usable. This weakeness highly limits the efficiency of relevant biomedical research. Over an extended period, tracking a larval zebrafish group is a labor-consuming manual work due to these systems' inability to generate reliable results.

It is small size, transparent and similar appearance, frequent occlusions, and discontinuous kinematics that combine to make larval tracking notoriously difficult [16]. Small size [see Fig. 1(a)] makes a multiple-sensor tracking system [26]–[28] and labeling approaches [29] infeasible due to imaging and operating constraints. Transparent and similar appearance [see Fig. 1(a)] causes tracking algorithms based on extracting superficial appearance features [30] to fail, as the difference between larvae's appearance is much less distinguishable than that of adult fish.

Occlusion is a general and the toughest tracking difficulty since occluded targets are recognized as a single connected component in image, and their identities may switch after the point of overlap [see Fig. 1(b)]. This issue can be corrected according to motion continuity. However, different from the adult's continual pattern of swimming, larvae can stay static over a long period and then flick, which is referred to as oscillatory movements [16]. This locomotive characteristic, along with larva's abruptly switched swim modes [7], [16], as shown in Fig. 1(c), yields to a long-lasting occlusion where larvae can change their motion stochastically. This results in the poor performance of systems that calculate the most likely assignment of identities before and after an occlusion based on continuous motion models [31], [32] and body geometry [33].

Presently, confronted with these tracking difficulties for larvae, the most widely used workaround is to segregate single larva in multiwell plates physically [34]. Another state-of-the-art solution, the well-known idTracker.ai [35], takes advantage of the powerful feature extracting capability of deep learning to identify objects individually. Besides, ZebraZoom [36] and Wang *et al.* [37] used background subtraction (BS) to segment and track larvae. Under certain circumstances, these methods work quite well, whereas it still seems that they have their limitations. Strictly limiting experiments to one zebrafish per dish constrains the research application as group behavior investigations, for example, cannot be performed. idTracker.ai is computationally and memory expensive and requires a whole recording to train the neural network, which disables this method to fit in real-time, streaming applications. ZebraZoom and the method proposed by Wang *et al.* is only capable of processing a restricted number of larvae (<10) as their capability to tackle the occlusion problem is limited.

We propose DanioSense (DS), an efficient, robust, and high-throughput tracking system, to automate the quantification of zebrafish larvae group movement. The task of tracking group larval zebrafish is to: 1) discover multiple objects in individual frames; 2) maintain the identity information across continuous frames; and 3) yield their trajectories while recording the statistical parameters [17], as shown in Fig. 2. The main contributions of this work include: 1) resolving the occlusion problem to obtain reliable trajectories with a light pretrained convolutional neural network (CNN) and a centerline extraction algorithm; 2) handling the discontinuous kinematics by an adaptive Kalman filter to obtain the optimal estimation of larvae's locomotive parameters; 3) adopting an enriched vector instead of a point to generate more detailed morphology and movement statistics; and 4) providing a variety of tracking visualization functions that offer users an intuitive sense of larval movements. In addition, DS is applicable

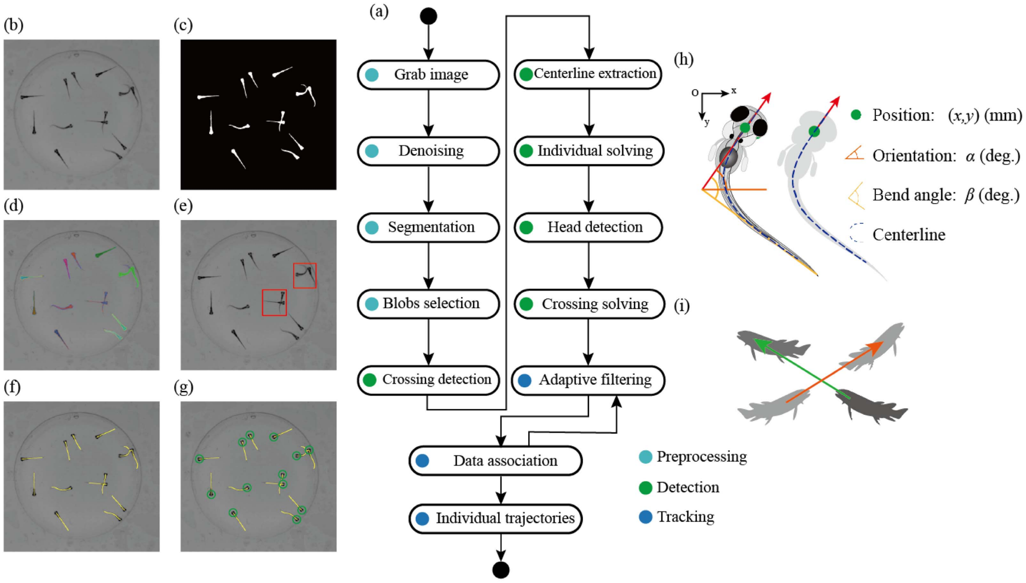

Fig. 3. Workflow of DS. (a) Schematic of the tracking algorithm for preprocessing, detection, and tracking. (b) Image input. (c) Segmentation. Extract blobs, a collection of connected pixels that are not part of the background, from a raw image, where potential objects are shown in white pixels, while the background is painted in black. (d) Blobs selection. Discard blobs that do not belong to the fish group according to their silhouettes. Each filtered blob is drawn in different colors. (e) Crossing detection. Distinguish individual objects from crossing objects. (f) Centerline extraction. Extract each object's "spine" to acquire its orientation and bend angle, printed in yellow lines. (g) Head detection. Since larvae's tail part is deformable, we track larvae's rigid head part. (h) Fish representation. A compact vector representation is used to describe a larva, including position, orientation, and bend angle. (i) Data association. Associate detections that belong to the same individual in successive frames according to their motion information.

to the behavior quantification of other fish-like animals, only requiring users to upload their own data sets and retrain the CNN classifier. Experimental results demonstrated that this tracker achieved an accuracy of over 97% and a processing speed of over 30 frames/s. Its reliability is comparable with idTracker.ai while achieving online tracking.

II. System Overview

A. System Setup

DS was built around a microscope (Nikon SMZ25) equipped with a motorized control of focusing. A camera (JAI, G0-5000C-USB, 2560 (width) × 2048 (height) pixels at 40 frames/s) was connected to the microscope to capture images. The transparent water tank was 35 mm × 30 mm in size, with water to a depth of 6 mm and a light source located below. This system's software was coded in Python. The computer hardware included a hexa-core Intel i7-9750H CPU, an Nvidia GTX1660Ti GPU, 8-GB RAM.

B. System Workflow

The proposed larvae tracking workflow is summarized in Fig. 3(a), composed of three stages: preprocessing, detection, and tracking. In the preprocessing stage, object silhouettes are extracted from the raw image [Fig. 3(c)] with intensity thresholding or BS plus thresholding, followed by removing noise and discarding spurious regions via denoising and the blob filtering step. In this way, the foreground is selected, covering the larvae in the field of vision (FOV) and containing all the necessary information such as texture and geometry for further processing.

From Fig. 3(h), we can see that a larva's body is articulated and consists of two parts: a rigid head part and a flexible tail part. During its swimming, its deformable posterior part entails larvae a large variety of morphology patterns [Fig. 1(c)], while its head segment presents little deformation. Thus, an enriched vector is adopted to represent a larva. The vector includes three tracking variables, namely, position, orientation, and bend angle, as shown in Fig. 3(h).

The tracking variables are obtained in the detection stage. By utilizing a light pretrained neural network and a centerline extraction method, larvae's head positions and orientations are derived even when they intersect with each other. This detection capacity endows the algorithm the ability to overcome the occlusion challenge radically. Furthermore, single and intersection cases are processed with two pipelines, where the computation burden is less in the first case to improve tracking efficiency. It is noteworthy that DS can output detailed motility parameters with the combination of the preprocessing step and the detection step rather than just position information with state-of-the-art object detection networks such as YOLO [38].

In practice, measurements are all relatively noisy compared to the ground-truth data. How to acquire the optimal estimations of locomotive variables, such as position and velocity, through noisy measurements is within the domain of filtering. In the tracking stage, an Adaptive Kalman Filter is employed to overcome the corruption of larvae's highly discontinuous acceleration pulses, which cause the dreadful outputs of conventional Kalman Filters. A two-stage data association scenario computes the correspondence between two consecutive frames with the foreground and the prediction of the adaptive Kalman filter [see Fig. 3(i)]. The associated measurements are then fed back to the adaptive filter to calculate the optimal estimations of locomotive variables.

III. Image Preprocessing

The mission of the preprocessing stage is first to denoise, then identify image pixels that belong to the tracked larvae (foreground) and reject pixels within the background, and finally assemble these foreground pixels into topologically connected regions that can be mapped to an individual or multiple objects. DS provides two techniques to achieve this goal:

1) *Intensity Thresholding:* The most straightforward method to extract the foreground is intensity thresholding, which is suitable for a grayscale image with a simple background. First, a Gaussian filter is employed to smooth the raw image, and then, the pixels within a user-defined intensity range are adopted as the foreground.
2) *BS:* For an input image with a relatively complex background or inhomogeneous illumination, simply thresholding an image is not sufficient to segment the foreground. A commonly adopted method to remedy this problem is BS plus intensity thresholding based on the difference between a pixel and the corresponding one in the background. An adaptive Gaussian mixture model is applied to estimate a consistent background model by incorporating a mixture of Gaussians [39], one per pixel, computed from 300 frames sampled at the beginning of the recording and updated periodically in the following frames.

The output of this segmentation procedure is a binary image [see Fig. 3(c)]. Morphological opening and erosion are further adopted to eliminate noise, such as bubbles, excretion, and small particles. Morphological dilation is then used to connect split areas into an entire region that represents single or multiple larvae.

False-positive regions, namely foreground regions that are mistaken as objects, are discarded according to their contour information. Blobs whose area is under a size threshold or the ratio of its perimeter to the area is not within a fish-like range are rejected. In addition, there may be broken blobs whose tail part is missed, which is caused by the low-intensity contrast between larvae's tail part and the water tank margin. The blobs whose centerline length is below a threshold are given a warning label, and their bend angles are not calculated in the next detection stage.

IV. Detection

DS is a detection-based tracking (DBT) paradigm, where trajectories are generated through detections. Therefore, to a great extent, the accuracy and robustness of detections determine the performance of tracking.

As previously mentioned, larvae's mutual occlusion and burst swimming pattern pose the main challenges. The most straightforward way to address them is to identify individuals during intersections. The underlying reasons for this choice are twofold. First, larvae's morphological and locomotive characteristics cause existing techniques to tackle occlusions seemingly impractical. For example, the diversity of intersection [Fig. 4(a)] causes the low accuracy of the merge–split approach [36] and its inability to cope with serious occlusion cases and its discontinuous kinematics leads to probabilistic methods' poor performance [16]. Second, an individual can modify its behavior drastically during and after an interaction [40], which contains valuable statistical information. Accordingly, identifying individuals during occlusions is not only crucial for tracking performance but also beneficial for interaction and group behavior research.

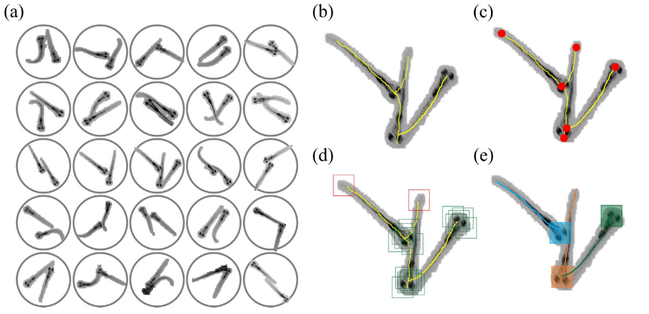

Fig. 4. Occlusion solving. (a) Miscellaneous crossing patterns. (b) Extracted centerline for an occlusion case of three larvae, painted in yellow. (c) Endpoints and intersections of the centerline plotted in red circles. These points are used to propose anchor windows. (d) Anchor windows used for localizing head position. The windows within the head region are illustrated in green rectangles, whereas others are in red ones. (e) Head detections and split centerlines assigned to each object. Individuals are described in different colors.

The core components in the detection stage include a centerline extraction module and a classification neural network. With the segmented regions from the preprocessing stage, the objects are grouped into single and multiple ones. Correspondingly, two pipelines are applied to process them for efficiency.

A. Crossing Detection

The first step in the detection stage is to detect occlusion events automatically. Let $B=\{b_1,\ldots,b_n\}$ be the collections of blobs segmented in the preprocessing stage and $A = \{\text{area}(b_i) \text{ for every } b_i \in B\}$ be the collection of the corresponding blob areas. The individual's area is defined by $m_A = \text{median}(A)$ and standard deviation $s_A = \sigma(A)$. If b is a blob, we define

$$\gamma(b) = \begin{cases} \text{is an individual,} & \text{if } |\text{area}(b) - m_A| < 4 \cdot s_A \\ \text{is a crossing,} & \text{otherwise.} \end{cases} \quad (1)$$

In this way, single and multiple blobs are grouped and processed by two different pipelines in the downstream steps.

B. Centerline Extraction

Centerline extraction is performed on the foreground blobs obtained in the preprocessing stage. Centerline, representing morphological skeleton, is the connected centerpoints of the maximum disks that locate within the contour, as shown in

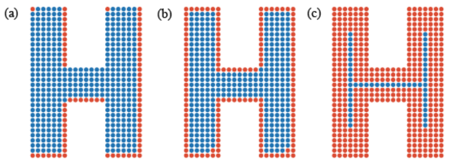

Fig. 5. Extract the centerline of the character H. Each point represents a pixel, and the blue point is the one to retain, whereas the orange one is removed. (a) First pipeline deletes the north–west corner points and the south–east boundary points. (b) Following the first pipeline, the second one deletes the south–east corner points and the north–west boundary points. (c) Extracted centerline of the character H. These two pipelines iterate alternately until no more pixels can be removed.

Fig. 3(h). The larva's centerline is utilized to localize the head point, determine the orientation, and calculate the bend angle.

A fast parallel thinning algorithm developed by Zhang and Suen is used to refine the centerline of the belt-like fish body [41]. This method consists of two pipelines: one deletes the north–west corner points and the south–east boundary points, whereas the other eliminates the south–east corner points and the north–west boundary points. These two parts iterate alternately until no more pixels can be removed. A brief illustration of the centerline extraction algorithm is shown in Fig. 5.

Fig. 4(b) shows the output of this centerline extraction method for an occlusion case of three larvae. Despite complex shape variations of individuals and occlusions [see Fig. 4(a)], centerline depicts the primary morphological structure of zebrafish within a blob. The endpoints and intersections of it [see Fig. 4(c)] indicate the possible locations of larval head and tail. This morphological information draws the attention of the CNN classifier and reduces the number of anchor windows to process in the head detection step.

C. Head Detection

A light pretrained CNN is applied to determine the head position. A single larva's head is localized by deciding which endpoint of its centerline is the head with the CNN, and its orientation is calculated accordingly. For mutual occlusion, a user-defined number of anchor windows are proposed, subject to a normal distribution with the endpoints and intersections of the centerline as the mean. Afterward, CNN is applied to categorize the windows into head windows and nonhead windows [see Fig. 4(d)]. The filtered head windows are further clustered to compute the head positions. This detection procedure also makes DS applicable to other fish-like animals with the substitution of CNN.

CNN is chosen as the classifier due to its excellent feature extraction capability, whose detailed architecture is shown in Fig. 6, consisting of two convolutional layers C1 and C2, two subsampling layers S1 and S2, and a full connection layer F1. The CNN's input is a gray image within the anchor window (40 (weight) × 40 (height) pixels), whose size is a bit larger than that of the larval head. This gray image is generated through the AND operation between the original gray image [see Fig. 3(b)] and the segmented image [see Fig. 3(c)]. The kernel size of C1 and C2 is 3×3 and 3×3 with the number of feature maps is 6 and 12, respectively. The activation function chosen for convolutional layers is ReLu. The number of feature maps of S1 and S2 equals 6 and 12, respectively. The S2 layer's output is reshaped, concatenated, and transferred to the full connection layer whose size is 108×1. Finally, a softmax function is applied to calculate the probabilities that the anchor window contains the fish head and does not contain the head. These two probabilities are the outputs. More details are shown in Section VII.

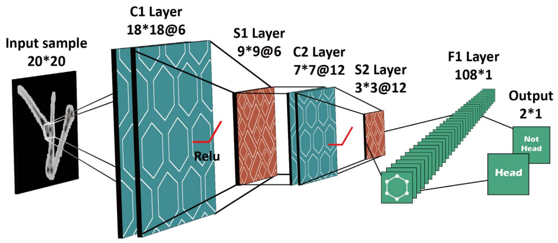

Fig. 6. CNN architecture used for classifying head and nonhead samples.

For occlusion, with the selected head windows, a mean-shift clustering method is further employed to cluster these windows and compute the mean of clustered points as the head position [42]. This approach automatically sets the number of clusters according to the density of samples without relying on the fixed inputting number of clusters adopted by methods, such as K-means clustering [43]. Crucially, it is robust to erroneous windows, if any, produced by the CNN classifier since a few outliers are grouped as the isolated points that can be filtered through a number threshold. Given the head points, the overlapped blob's centerline is partitioned to calculate each individual's orientation [see Fig. 4(e)].

V. Tracking

From a probabilistic inference perspective, object tracking is a multivariable estimation problem-estimate the optimal sequential states of all the objects given the measurements. This task consists of two parts: 1) associating measurements of the present frame with the estimations of the previous frame and 2) calculating the optimal estimations of the present frame through noisy measurements.

The tracking stage follows a predict-update cycle. In the first frame, for each object, a corresponding adaptive Kalman filter is initialized and uses the measurement as its estimation. After initialization, first, each object's adaptive Kalman filter in the present frame makes a prediction for the next frame. Afterward, the data association protocol links the current frame estimations with the next frame measurements from the detection stage. Finally, if the filter receives an associated measurement, it outputs the optimal estimations for the next frame, integrating the prediction and the measurement; otherwise, it adopts the current frame prediction as the temporal next frame estimation for the next data association.

A. Adaptive Kalman Filter

It is based on the conventional Kalman filter. The typical workflow of Kalman filter for tracking can be expressed as

Prediction:

$$\hat{x}_k = F x_{k-1} \tag{2}$$

$$\hat{P}_k = F P_{k-1} F^T + Q_k \tag{3}$$

Innovation:

$$y_k = z_k - H\hat{x}_k \tag{4}$$
$$S_k = H\hat{P}_k H^T + R_k \tag{5}$$

Updating:

$$K_k = \hat{P}_k H^T S_k^{-1} \tag{6}$$
$$x_k = \hat{x}_k + K_k y_k \tag{7}$$
$$P_k = (I - K_k H)\hat{P}_k \tag{8}$$

where $\hat{x}_k$ is the predicted state vector, $\hat{P}_k$ represents the covariance matrix for $\hat{x}_k$, F demonstrates the transition matrix, Q_k is the process noise covariance matrix, y_k indicates the innovation, S_k is the covariance matrix for y_k, z_k denotes the measurement, R_k demonstrates the measurement noise covariance matrix in the kth frame, K_k represents the Kalman gain, x_k indicates the estimation of filtering, P_k is the covariance matrix for x_k, and the subscript k denotes the kth frame.

The most widely used process model is the constant velocity (CV) model, which works well for continuous swimming but is problematic when applied to quickly maneuvering objects such as larvae. They can remain stationary over a long period and then flick, which causes the velocity estimation to be fairly small in the static stage and lag when burst swimming happens. Conversely, the constant acceleration (CA) motion model can track the burst swimming well but mistake noise as accelerations even when larvae show little or no movement.

Integrating a CV model and a CA model, an adaptive Kalman filter is proposed. When a maneuver happens, the process noise covariance matrix Q_k^v of the CV model is adjusted to help the estimation of the CV model converge [44], and the outputs of these two models are blended by attaching higher importance to the CA model to track the maneuver. Once the object returns to a static mode, the CV model is given more weight for a smooth estimation.

The CV model's state variable x_k^v of the larva at frame k is modeled as

$$x_k^v = [x^v, y^v, \theta^v, \dot{x}^v, \dot{y}^v, \dot{\theta}^v]^{\mathrm{T}} \tag{9}$$

where (x^v, y^v) is the head position of the larva, θ^v is the orientation, and the superscript v denotes that these variables are from the CV model. Similarly, The CA model's state variable x_k^a is defined as

$$x_k^a = [x^a, y^a, \theta^a, \dot{x}^a, \dot{y}^a, \dot{\theta}^a, \ddot{x}^a, \ddot{y}^a, \ddot{\theta}^a]^{\mathrm{T}}. \tag{10}$$

The state-transition matrix F^v and the measurement matrix H^v for the CV model are given by

$$F^v = \begin{bmatrix} I_3 & I_3 T \\ 0_3 & I_3 \end{bmatrix}, \quad H^v = [I_3 \quad 0_3] \tag{11}$$

where T is the sampling interval, I_3 and 0_3 are 3×3 identity and zero matrices, respectively, and F^a and H^a for the CA model are defined by

$$F^a = \begin{bmatrix} I_3 & I_3 T & 0.5 I_3 T^2 \\ 0_3 & I_3 & I_3 T \\ 0_3 & 0_3 & I_3 \end{bmatrix}, \quad H^a = [I_3 \quad 0_3 \quad 0_3]. \tag{12}$$

A maneuver is detected according to the normalized square of the residual of the CV model

$$\varepsilon_k^v = \left(y_k^v\right)^T \left(S_k^v\right)^{-1} y_k^v \tag{13}$$

where y_k^v is the innovation for the CV model, S_k^v is the covariance matrix for y_k^v, and ε_k^v depicts the distance between the measurement and the prediction of the CV model. If ε_k^v exceeds a user-defined threshold $\varepsilon_{\max}$, it means that a burst movement happens. The process noise matrix Q_k^v indicates how accurate the CV model's prediction x_k^v is. A large ε_k^v means that x_k^v is inaccurate and unreliable; in other words, Q_k^v is too small. Q_k^v is adjusted to enable the CV model to catch up with the maneuver, using

$$\begin{cases} Q_{k+1}^v = \alpha Q_k^v, \ c = c+1, & \text{if } \varepsilon_k^v > \varepsilon_{\max} \\ Q_{k+1}^v = Q_k^v/\alpha, \ c = c-1, & \text{if } \varepsilon_k^v < \varepsilon_{\max} \text{ and } c > 0 \\ Q_{k+1}^v = Q_k^v, & \text{if } \varepsilon_k^v < \varepsilon_{\max} \text{ and } c = 0 \end{cases} \tag{14}$$

where α is the coefficient to change the value of Q_k^v and c is a variable starting with the value of 0 to count the total times Q_k^v is adjusted. The coefficient α is defined by users, whose value aligns with the locomotivity of objects, e.g., 10 for passive larvae and 100 for proactive ones. The value of α determines how fast the CV model can catch up with the burst movement, but it does not affect the process of tracking the maneuver. Maneuver tracking is achieved by assigning a higher weight to the CA model. The estimation of the adaptive Kalman filter x_k combines the outputs of the CV model x_k^v and the CA model x_k^a, according to

$$x_k = p_k^v \begin{bmatrix} x_k^v \\ 0_{1\times 3} \end{bmatrix} + p_k^a x_k^a \tag{15}$$

where $0_{1\times 3}$ is a 1×3 zero vector and p_k^v and p_k^a are the weights assigned to these two models and they are calculated by

$$p_k^v = \frac{\mathcal{L}_k^v p_{k-1}^v}{\mathcal{L}_k^v p_{k-1}^v + \mathcal{L}_k^a p_{k-1}^a} \tag{16}$$
$$p_k^a = \frac{\mathcal{L}_k^a p_{k-1}^a}{\mathcal{L}_k^v p_{k-1}^v + \mathcal{L}_k^a p_{k-1}^a} \tag{17}$$

where $\mathcal{L}_k^v$ and $\mathcal{L}_k^a$ are the likelihood of the CV and CA model, respectively, and are utilized to describe how likely a filter is to be performing optimally given the inputs. The likelihood $\mathcal{L}_k$ is defined as

$$\mathcal{L}_k = \frac{1}{\sqrt{2\pi S_k}} \exp\left(-\frac{1}{2} y_k^{\mathrm{T}} S_k^{-1} y_k\right). \tag{18}$$

Once a maneuver is detected, a higher value $\mathrm{w}_{\max}^a$ is given to p_{k-1}^a to track the burst movement closely. When $\varepsilon_k^v < \varepsilon_{\max}$ and $c = 0$, p_{k-1}^a returns to the initial value $\mathrm{w}_{\min}^a$ to reject the measurement noise. This process is shown in

$$\begin{cases} p_k^a = \dfrac{L_k^a \mathrm{w}_{\max}^a}{L_k^v\left(1 - \mathrm{w}_{\max}^a\right) + L_k^a \mathrm{w}_{\max}^a}, & \text{if } \varepsilon_k^v > \varepsilon_{\max} \text{ and } c = 0 \\ p_k^a = \dfrac{L_k^a \mathrm{w}_{\min}^a}{L_k^v\left(1 - \mathrm{w}_{\min}^a\right) + L_k^a \mathrm{w}_{\min}^a}, & \text{if } \varepsilon_k^v < \varepsilon_{\max} \text{ and } c = 0 \\ p_k^a = \dfrac{L_k^a p_{k-1}^a}{L_k^v p_{k-1}^v + L_k^a p_{k-1}^a}, & \text{otherwise} \end{cases} \tag{19}$$

while $p_k^v = 1 - p_k^a$. Jagged switches are avoided because $\mathcal{L}_k^v$ and $\mathcal{L}_k^a$ are considered after the mandatory value assignment to p_{k-1}^a.

B. Data Association

Let $X_{k-1} = (x_{k-1}^1, x_{k-1}^2, \ldots, x_{k-1}^{M_{k-1}})$ denote the estimations of all the M_{k-1} objects in the $(k-1)$th frame and $Z_k = (z_k^1, z_k^2, \ldots, z_k^{N_k})$ denote the measurements for all the detected N_k objects in the kth frame. The objective of data association is to link x_{k-1}^i and z_k^j that belong to the same object, in other words, associate X_{k-1} with Z_k. With sufficient movement information from the detection stage, the trajectory of each identified larva is generated in this step. A two-stage association method is designed to reduce the time complexity and improve the robustness.

Let $B_k = (b_k^1, b_k^2, \ldots, b_k^{N_k})$ be the collected blobs for all the detected N_k objects in the kth frame from the preprocessing stage. It is obvious that b_{k-1}^i and b_k^j belong to the same object if the intersection of these two sets of pixels is very large. The intersection over union (IOU) is adopted to measure this, which is defined as

$$\mathrm{IOU}(i, j) = \frac{b_{k-1}^i \cap b_k^j}{b_{k-1}^i \cup b_k^j} \tag{20}$$

where $b_{k-1}^i \cap b_k^j$ is the number of pixels that b_{k-1}^i and b_k^j share and $b_{k-1}^i \cup b_k^j$ is the number of pixels that b_{k-1}^i and b_k^j cover in the frame. Following the crossing detection step, an AND operation is performed between the foreground of the $(k-1)$th frame and the kth frame, by which the intersection area is extracted. For each overlapped region $b_{k-1}^i \cap b_k^j$ that is part of the single larva, these two blobs are linked if the $IOU(i, j)$ is beyond a threshold. In this way, a good few single objects are associated.

Assuming that C_k is the number of associated objects in the previous step, A $(M_{k-1} - C_k) \times (N_k - C_k)$ cost matrix $D(X_{k-1}, Z_k)$ is initialized to associate the remnant X_{k-1} with the left Z_k

$$D(X_{k-1}, Z_k) = \begin{bmatrix} d_{x_{k-1}^1, z_k^1} & \cdots & d_{x_{k-1}^1, z_k^{N_k - C_k}} \\ d_{x_{k-1}^2, z_k^1} & \cdots & d_{x_{k-1}^2, z_k^{N_k - C_k}} \\ \vdots & \ddots & \vdots \\ d_{x_{k-1}^{M_{k-1} - C_k}, z_k^1} & \cdots & d_{x_{k-1}^{M_{k-1} - C_k}, z_k^{N_k - C_k}} \end{bmatrix} \tag{21}$$

where

$$d_{x_{k-1}^i, z_k^j} = p_{k-1}^v \sqrt{\left(z_k^j - H^v \hat{x}_k^v\right)^{\mathrm{T}} \left(S_k^v\right)^{-1} \left(z_k^j - H^v \hat{x}_k^v\right)} + p_{k-1}^a \sqrt{\left(z_k^j - H^a \hat{x}_k^a\right)^{\mathrm{T}} \left(S_k^a\right)^{-1} \left(z_k^j - H^a \hat{x}_k^a\right)} \tag{22}$$

is the weighted Mahalanobis distance between x_{k-1}^i and z_k^j. p_{k-1}^v, H^v, S_k^v, and $\hat{x}_k^v$, and p_{k-1}^a, H^a, S_k^a, and $\hat{x}_k^a$ are the likelihood, the measurement matrix, the covariance matrix, and the prediction of the CV model and the CA model of the adaptive Kalman filter for the ith object, respectively. The Mahalanobis distance is employed to measure the distance between the estimation of the $(k-1)$th frame and the measurement of the kth frame. It takes the uncertainty of each object's states, as well as correlations of each state (e.g., coordinate x and coordinate y), into account through the covariance matrix S_k of the corresponding Kalman filter. If S_k^v and S_k^a are both a diagonal matrix, which means that there is no correlation between each state, the weighted Mahalanobis distance $d_{x_{k-1}^i, z_k^j}$ equals the weighted scaled Euclidean distance.

The optimal Hungarian algorithm [45] is employed to match the remnant objects. This algorithm is a greedy bipartite graph matching method to assign a unique z_k^j to x_{k-1}^i according to the cost matrix $D(X_{k-1}, Z_k)$. Afterward, a validation step is taken. Assignments fall within a gating threshold, namely within the movement capabilities of larvae, are adopted by the adaptive Kalman filter to get the optimal estimation of states, while the erroneous links are rejected. For unassigned objects and those whose assignments are discarded, the predictions of corresponding filters are adopted as the estimated states in the current frame.

The data association step can also be carried out by solving the cost matrix $D(X_{k-1}, Z_k)$ alone. However, the time complexity of the optimal Hungarian algorithm is $\mathrm{O}(\mathrm{n}^3)$, and the calculation of the weighted Mahalanobis distance $d_{x_{k-1}^i, z_k^j}$ involves the matrix computation. When the number of objects multiplies, the computation burden increases significantly, further undermining the tracking efficiency. Nevertheless, the IOU calculation only requires the AND operation between two images, which is hardly affected by the number of objects. By filtering quite a few single objects in advance, the computation burden is remarkably alleviated.

VI. Experimental Results and Discussion

We captured five data sets (denoted as D1–D5) to evaluate the performance of the proposed method. Each data set consisted of 4000 frames, where the sizes of each zebrafish larvae group aged between three and five days postfertilization (dpf) were 5, 10, 20, 30, and 40, respectively. It is noteworthy that DS's tracking throughput is subject to DS's imaging equipment, which can be further enhanced with a microscope that has a larger visual field. Meanwhile, it should be ensured that the camera can output an image to be clear enough for CNN to identify objects (at least around 2000 pixels per object). With the system configuration in Section II-A, the object density of 40 larvae is sufficient to evaluate the capability of DS to tackle occlusions, shown in the Supplementary Video.

We adopted six performance evaluation metrics to assess the detection and tracking performance [46], including identification rate (IR), false positives (FPs), fragments (Frag), ID switches (IDSs), multiple-object tracking accuracy (MOTA), and processing speed (Hz), as shown in Table I.

A. Performance of Detection

DS starts with the preprocessing stage to extract blobs and proceeds to the detection stage to determine the tracking parameters. We randomly sampled 500 frames from the data sets and labeled each larva manually to evaluate the performance.

IR for head position and orientation were both 97.6%. Missing detections were caused by the occlusion of transparent larval body, where the body was severely occluded by other objects or the tank shadow. With the integration of the CNN

TABLE I

DESCRIPTION OF THE PERFORMANCE METRICS

Metrics	Description
IR	The ratio of a tracking parameter that is successfully identified, including IR for position (error$<200\mu$m), IR for orientation (error$<20°$), and IR for bend angle (error$<20°$).
FP	The number of false positives, namely, the number of the measurement that does not correspond to any object.
Frag	The number of times a trajectory is interrupted, which happens when no measurement is assigned to an object.
IDS	The number of identity switches, which occurs when the tracker assigns a wrong measurement to an object.
MOTA	1-(FP+Frag+IDS)/(the total number of ground truth frames)
Hz	The number of frames processed per second.

classifier and mean-shift clustering, the unoccluded head parts were successfully localized. The detection accuracy was high, with a mean error of 102 μm for position and a mean error of 8.8° for orientation, as shown in Fig. 7(a); 30% of the positions and angles are within the yellow region's contour, whereas 75% and 97% of them are within the contour of green and blue regions, respectively. Overall, nearly, all the detections are within a reasonable region. The few erroneous detections are discarded temporally in the tracking stage because the association cost is too large. It should be noted that the distribution and accuracy of detections are subject to the image resolution, the classifying ability of CNN, and the number of anchor windows in the detection stage, which means that the accuracy can be further enhanced by improving them.

The IR for bend angle was 91.1%. The rest 8.9% unidentified bend angles resulted from broken blobs and severely overlapped objects, which caused the centerline to be relatively short. For objects whose centerline is under a length threshold, DS attaches a warning label to their bend angle for correction in further statistics.

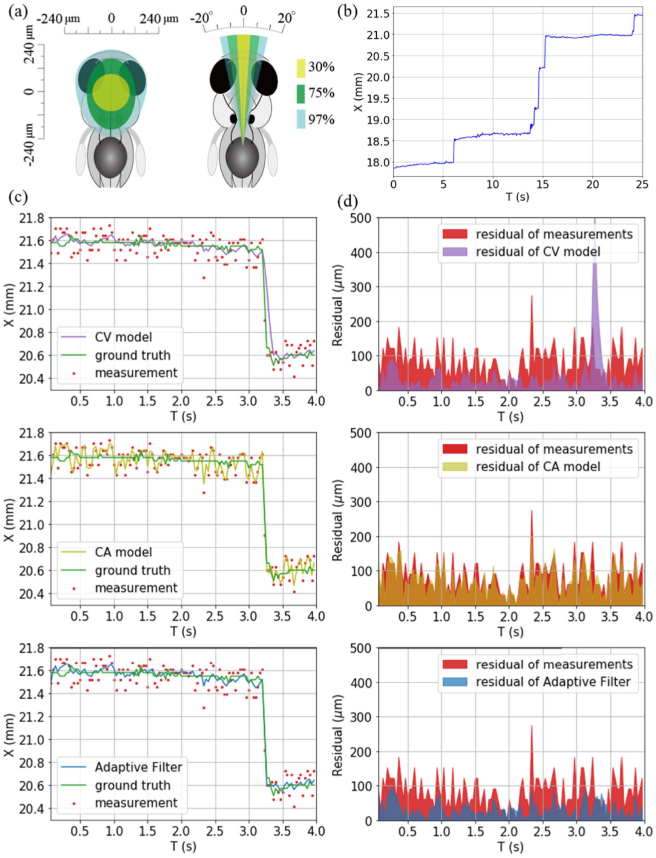

Fig. 7. (a) Accuracy of position and orientation. Distributions of them are plotted in different colors. For example, 30% of all the positions and angles are within the yellow region's contour. (b) Fragment of a trajectory's x coordinate from a randomly selected larva. (c) Estimated coordinate x of a trajectory with the conventional Kalman filter using a CV model, the conventional Kalman filter using a CA model, and the adaptive Kalman filter versus measurements and ground-truth x coordinates, respectively. (d) Absolute residual errors with CV model, CA model, and adaptive Kalman filter.

B. Performance of Tracking

A trial of 200 frames was selected from each data set at a random start, for a total of 1000 frames. Each trail was manually annotated to judge the tracking performance, which is referred to as ground truth. The underlying evaluation criteria are twofold: how optimally the tracker can estimate the real states despite the measurement noise and abrupt motions and how robustly the tracker can maintain the identities of each individual across occlusions.

As shown in Fig. 7(b), larvae's movement presented highly discontinuous acceleration pulses, where freezing motion segments were interrupted by disruptive transitions. Compared with the suboptimal performance of conventional Kalman filters, the adaptive Kalman filter significantly reduced the measurement noise that was mistaken by the CA model as accelerations and adapted itself to track the abrupt dynamic closely, in which the CV model cannot accomplish, as shown in Fig. 7(c) and (d). The introduction of adaptive filtering achieves optimal quantification of motility parameters.

The tracking performance of DS was compared with two other methods, namely BS and simplified DanioSense (SDS). The first method was proposed by Wang *et al.* [37], which is corresponding to the image preprocessing stage in DS. The second method was the SDS, modeling the larva as a point instead of the enriched representation, whereas the handling of occlusion is almost the same. The detailed evaluation results were listed in Table II.

According to the experiment results in Table II, we can see that occlusion plays a pivotal role in the tracking performance. When the group size was limited, BS performed well even if it did not take occlusion into account. However, with the growth of larval group size, the frequency of occlusion increased, and the performance of BS dropped significantly.

SDS was able to assign the correct identity to each object in standard occlusion cases without considering orientation [see Fig. 8(a)]. However, considering extreme cases in Fig. 8(b), where the heads of two objects overlapped, with one turned extremely fast and another was close, it would be difficult to maintain correct identities by utilizing SDS since the point representation associated objects only by calculating the distance between them. As the number of objects increased, the serious occlusion events happened more frequently and led to the performance drop of SDS. In this situation, the inclusion of orientation endowed DS the capability to generate trajectories with nearly no IDSs whatever the fish group size

TABLE II
PERFORMANCE OF LARVAE TRACKING

Data	Method	FP	Frag	IDS	MOTA(%)	Hz(fps)
D1	BS	0	16	0	99.2	98.4
	SDS	0	0	0	100	50.6
	DS	**0**	**0**	**0**	**100**	**45.4**
D2	BS	0	103	0	94.6	98.4
	SDS	0	12	0	99.4	47.6
	DS	**0**	**12**	**0**	**99.4**	**43.1**
D3	BS	0	>600	>500	<72.5	97.1
	SDS	0	54	0	98.7	38.1
	DS	**0**	**54**	**0**	**98.7**	**37.5**
D4	BS	0	>700	>600	<78.3	90.1
	SDS	2	76	124	96.6	33.9
	DS	**2**	**76**	**0**	**98.7**	**33.6**
D5	BS	0	>1100	>2800	<45.8	82.3
	SDS	0	229	219	94.7	31.9
	DS	**0**	**179**	**60**	**97.0**	**31.7**

was. The principal factors that caused the IDS and Frag of DS included the low-intensity contrast between the transparent larval body and the dark water tank margin and the blurry appearance features when larvae flicked, which disabled DS to identify objects (see the Supplementary Video).

idTracker.ai was tested on large collectives of juvenile zebrafish (100 dpf) in [35], giving accuracies of 99.96% ± 0.06% for 60 individuals and 99.99% ± 0.01% for 100 individuals. Along with its excellent performance comes the cost of efficiency and heavy computation burden. For a video clip of 60 zebrafish with a duration of 10’29” and file size of 228 Gb, it took over 5 h to track with a workstation (Nvidia TITAN X or GTX1080Ti GPU, 32Gb-128Gb RAM). The exorbitant hardware requirement and prolonged processing time limit its application.

idTracker.ai has two competitive edges over DS. Once the time-consuming training step is accomplished, idTracker.ai can identify each individual and associate them regardless of the experiment time and environment. Another advantage is that it is applicable to a broader range of animals, not just fish-like animals for DS. However, this pattern recognition approach also has a few limitations compared with DS. First, it requires long video sequences to acquire enough reference frames to train the neural network. The training process is computationally and memory expensive, which leads to a significantly low tracking efficiency. Second, it mainly uses interpolations to estimate objects’ positions when they overlap, creating many trajectory gaps (Fragment) when applied to zebrafish larvae.

Identifying head points during intersections, DS generated trajectories with few gaps; in other words, the value of fragments was remarkably low. The processing speed of DS was consistently over 30 frames/s even when the computation burden multiplied as the number of larvae increased. Multithreading contributes to the system’s high efficiency, where each object is localized and tracked parallelly. The processing speed can be further improved with better hardware (Intel i7-9750H CPU, Nvidia GTX1660Ti GPU, 8-GB RAM in the experiment) and software framework optimization. With the number of objects multiplying, the processing speed drops relatively, and the upper limit is no more than that of the method BS (about 60 frames/s for 100 objects with the hardware in this article). In addition, DS can also run on an ordinary computer with no GPU at a speed of about 1 frame/s, which removes the barrier for users to have access to a reliable automatic tracker for a larval group.

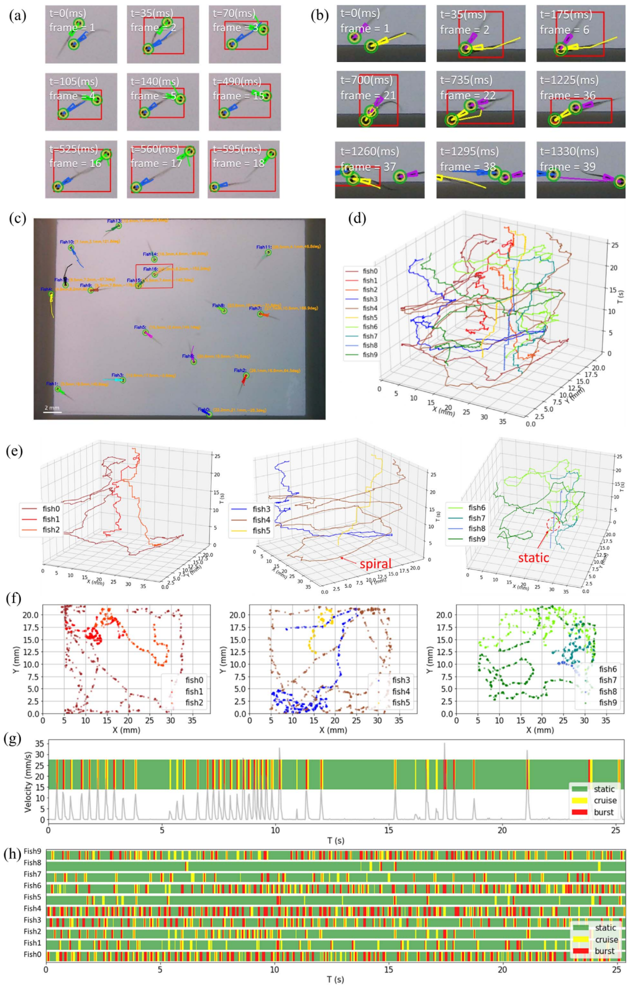

Fig. 8. (a) Occlusion solving for standard cases by DS. Green circles indicate the detected head positions. Orientations are shown in triangles with the corresponding colors for each object. In this case, each larva’s head part keeps away from each other, which is relatively easy to handle. (b) Occlusion solving for severe cases by DS, where larvae’s heads overlap. IDSs may happen if the orientation is not considered. (c) Tracking outputs of DS. Individuals are shown as colored triangles with a history tail of ten latest frames, and head positions and orientations are printed near the corresponding objects. Red rectangles represent occlusion cases. (d) Resultant trajectories of ten larvae. Z-axis indicates time, X-axis and Y-axis are coordinates of the image plane, and different colors represent the corresponding individuals. (e) Resultant trajectories of ten larvae illustrated in three subfigures. Users can adjust the viewpoint to observe the movement patterns of larvae. For example, the spiraling trajectory of fish 4 indicates that it tends to move in circles, whereas the vertical trajectory of fish 8 means that it is inactive. The trajectories of fishes 4 and 8 are highlighted with red arrows and circles. (f) Scatter plots of ten larvae’s positions for 26 s. The color indicates identity; higher opacity represents higher positional preference. (g) Velocity of one object over 25 s and its behavioral ethogram. (h) Behavioral ethograms for ten larvae, visualizing the dynamic differences between them.

C. Tracking Visualization

As shown in Fig. 8(c), DS prints the identity and tracking parameters near the corresponding object and automatically labels the occlusion events in real time. This function serves as a behavior detector where users can specify desired quantitative data and motor patterns to be presented rather than estimating them subjectively. The behavior detector brings efficiency to behavioral science investigations [47], such as the identification of escape response, optokinetic reflex, and behavioral abnormality.

Acquisition of complete trajectories despite occlusions enables users to analyze motor patterns for individuals and further social interactions within a group. Each object's trajectory is plotted in different colors in Fig. 8(d) where their spatial and temporal relationships are presented. DS provides the function for users to select the trajectory to print and adjust the viewpoint of the figure to observe the movement pattern of the selected larva, as shown in Fig. 8(e). The position change of objects with time cannot be viewed through a planar figure like Fig. 8(f), which visualizes larval positions' spatial distributions during social interactions. In addition, DS can plot the tracking results of the assigned frame by users in the form of Fig. 8(c). Furthermore, DS generates each object's ethogram [25], which is an individual's quantitative behavioral description by segmenting the trajectory into bouts, where the object performs a given behavior. In this way, the trajectory is condensed into meaningful and quantitative statistics.

Larval behavior patterns in the experiment were simply categorized into three types, namely, static, cruise, and burst, according to the amplitude of the velocity and the normalized square of the residual of the adaptive Kalman filter ε_k^v (static: velocity < 3 mm/s and $\varepsilon_k^v < \varepsilon_{\max}$, cruise: velocity > 3 mm/s and $\varepsilon_k^v < \varepsilon_{\max}$, and burst: velocity > 3 mm/s and $\varepsilon_k^v > \varepsilon_{\max}$), as shown in Fig. 8(g). Ethograms for ten larvae are shown in Fig. 8(h), which visualizes the activeness of each object and the difference among them. As we can see, the proportion and frequency of burst episodes vary from individual to individual. The distribution of behavior patterns of each object over time is clearly presented in this chart. Large-scale chemical and genetic screens would benefit from this visualization function a lot, where operators can gain insight into the underlying mechanisms through tedious numerical data.

All of these charts can be obtained in real time during online tracking, offering users an intuitive sense of larval locomotion and significantly boosting the efficiency of relevant biomedical research. For example, the mechanism and evolution of collective behavior reveal the underlying neuronal circuits and molecular pathways in neuroscience [7] and also unearths the social hierarchies within a larval group [18]. Due to larvae's genetic similarity with human [6], the study of molecular and neuronal principles of social interactions also facilitates the understanding and treatment of social-related symptoms, such as autism and schizophrenia [48].

D. Discussion

DS can also be applied to other fish-like species, during which several factors need to be considered. First, the intensity contrast between objects and background should be high enough to enable the image processing stage to segment blobs. If possible, the dark water tank margin should be avoided because objects are blocked. Supposing that an object is not detected for an extended period, its current ID is deleted, and it will be assigned a new ID the next time it reappears, which causes an entire trajectory to be divided into many short segments (tracklets). In this case, a postprocessing step is used to associate these segments belonging to the same object into an entire one. DS's segmentation methods are also applicable to complex static backgrounds or those changing slowly. However, they are not suitable for fast-changing ones. To tackle this, users can apply a state-of-the-art image segmentation neural network instead of the original methods of DS but at a higher computation cost. DS's tracking modules are flexible, and users can also customize their functions to replace the original ones.

When tracking a new species, the CNN classifier needs to be retrained. DS provides a semiautomatic function to assist users in acquiring their customized data set, described in Section VII. In practice, the classification accuracy, the ratio of the number of correctly classified samples to the total number, should be at least 90% to guarantee a feasible detection accuracy. For a higher detection accuracy, image resolution improvement, a deeper CNN structure, and a higher number of anchor windows can be adopted, which is a tradeoff between efficiency and accuracy.

The parameters of the adaptive Kalman filter require adjustment according to the dynamic features of different animals. The threshold $\varepsilon_{\max}$ is used to determine whether a burst movement happens; for a smoothly swimming fish, it can be large, which means that there is no burst movement. The measurement noise covariance matrix R_k for CA and CV models should align with the detection accuracy. The process noise covariance matrix Q_k for CA and CV models can be given a bit higher value for more energetic objects. Users can also adjust other parameters a bit, but it does not make a big difference.

With DS, individual locomotive characteristics and trajectories within an animal group can be obtained in real time with high accuracy and strong robustness. With the abundant movement information, more comprehensive criteria for quantifying different animals' individual and group behavior can be defined, which is further used for tracking visualization. In some applications [16], more detailed motion information needs to be obtained with a high-speed camera at a speed of over 500 frames/s. Therefore, enhancing the processing efficiency is one essential step.

VII. Conclusion

Zebrafish larva is an ideal vertebrate model organism for large-scale chemical and genetic screens. This article proposed DS, an automatic, robust, and high-throughput tracking system for a zebrafish larval group. In the previous research, the behavior observation of zebrafish larvae is mainly on the individual level, where the screening throughput is limited by mechanical devices, and group behavior cannot be studied. The key step of DS is the resolution of the occlusion problem, which enables high-throughput group tracking and further locomotive statistics extraction. The combination of CNN and centerline extraction endows DS the capability to identify objects in occlusions, which is a major advance over existing methods. The introduction of adaptive Kalman filter achieves optimal quantification of motility parameters for further analysis. Based on the reliable movement information, a wide range of visualization options is provided to bring insight into underlying patterns through massive amounts of data. Integrating these techniques, DS, the next steps include further improvement in processing efficiency and extension of applications.

Appendix
Further Details for CNN Training

The mini-batch stochastic gradient descent strategy is employed to train the CNN [49], with a data set consisting of 5000 head images (40 (weight) × 40 (height) pixels) and 7082 nonhead images (40 (weight) × 40 (height) pixels). The whole data set was divided into three sections: 70% for training, 25% for validation, and 5% for test. An image is classified as head if the probability for the head image is over 0.9. The CNN achieved a classification accuracy of 95.2% in the training stage, 97.6% in the validation stage, and 99.8% in the test stage. Practically, a few classification errors are acceptable and beneficial. They can be corrected in the detection stage, where the mean-shift clustering isolates and discards them. Furthermore, they help avoid overfitting and contribute to the generalization and robustness of the classifier.

Daniosense adopts a semiautomatic strategy to make the data set. First, 12 082 images (40 (weight) × 40 (height) pixels) are sampled automatically with the proposed anchor windows in Section IV-C. Then, the operator labels 2000 of them manually, distinguishing head images from nonhead images. The labeled 2000 images are used to train the first CNN. Afterward, the first CNN is used to classify another 4000 images and add labels to them, after which the operator checks and corrects the labels. With the newly labeled 4000 images and the first 2000 ones, the first CNN is further trained to the second CNN. After a couple of rounds, users can get a CNN with satisfactory classification accuracy. Users can also define the number of images for labeling in each round.

References

[1] O. Ronneberger *et al.*, "ViBE-Z: A framework for 3D virtual colocalization analysis in zebrafish larval brains," *Nature Methods*, vol. 9, no. 7, pp. 735–742, Jul. 2012.

[2] J. Rihel *et al.*, "Zebrafish behavioral profiling links drugs to biological targets and rest/wake regulation," *Science*, vol. 327, no. 5963, pp. 348–351, Jan. 2010.

[3] B. Fraysse, R. Mons, and J. Garric, "Development of a zebrafish 4-day embryo-larval bioassay to assess toxicity of chemicals," *Ecotoxicology Environ. Saf.*, vol. 63, no. 2, pp. 253–267, Feb. 2006.

[4] C. Saverino and R. Gerlai, "The social zebrafish: Behavioral responses to conspecific, heterospecific, and computer animated fish," *Behavioural Brain Res.*, vol. 191, no. 1, pp. 77–87, Aug. 2008.

[5] H. W. Detrich, III, L. Zon, and M. Westerfield, *The Zebrafish: Disease Models and Chemical Screens*. New York, NY, USA: Academic, 2011.

[6] K. Howe *et al.*, "The zebrafish reference genome sequence and its relationship to the human genome," *Nature*, vol. 496, no. 7446, pp. 498–503, 2013.

[7] A. V. Kalueff and J. M. Cachat, *Zebrafish Models in Neurobehavioral Research*. Cham, Switzerland: Springer, 2011.

[8] W. Wang, X. Liu, D. Gelinas, B. Ciruna, and Y. Sun, "A fully automated robotic system for microinjection of zebrafish embryos," *PLoS ONE*, vol. 2, no. 9, p. e862, Sep. 2007.

[9] W. H. Wang, X. Y. Liu, and Y. Sun, "High-throughput automated injection of individual biological cells," *IEEE Trans. Autom. Sci. Eng.*, vol. 6, no. 2, pp. 209–219, Apr. 2009.

[10] X. Zhang, Z. Lu, D. Gelinas, B. Ciruna, and Y. Sun, "Batch transfer of zebrafish embryos into multiwell plates," *IEEE Trans. Autom. Sci. Eng.*, vol. 8, no. 3, pp. 625–632, Jul. 2011.

[11] S. Zhuang, W. Lin, H. Gao, X. Shang, and L. Li, "Visual servoed zebrafish larva heart microinjection system," *IEEE Trans. Ind. Electron.*, vol. 64, no. 5, pp. 3727–3736, May 2017.

[12] G. Zhang *et al.*, "An integrated microfluidic system for zebrafish larva organs injection," in *Proc. 43rd Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Oct. 2017, pp. 8563–8566.

[13] S. Zhuang *et al.*, "Visual servoed three-dimensional rotation control in zebrafish larva heart microinjection system," *IEEE Trans. Biomed. Eng.*, vol. 65, no. 1, pp. 64–73, Jan. 2018.

[14] G. Zhang *et al.*, "Zebrafish larva orientation and smooth aspiration control for microinjection," *IEEE Trans. Biomed. Eng.*, vol. 68, no. 1, pp. 47–55, Jan. 2021.

[15] G. Zhang *et al.*, "Visual-based contact detection for automated zebrafish larva heart microinjection," *IEEE Trans. Autom. Sci. Eng.*, early access, Sep. 17, 2020, doi: 10.1109/TASE.2020.3019782.

[16] P. R. Martineau and P. Mourrain, "Tracking zebrafish larvae in group–status and perspectives," *Methods*, vol. 62, no. 3, pp. 292–303, Aug. 2013.

[17] S. A. Budick and D. M. O'Malley, "Locomotor repertoire of the larval zebrafish: Swimming, turning and prey capture," *J. Exp. Biol.*, vol. 203, no. 17, pp. 2565–2579, 2000.

[18] D. J. Sumpter, *Collective Animal Behavior*. Princeton, NJ, USA: Princeton Univ. Press, 2010.

[19] C. Milanese *et al.*, "Hypokinesia and reduced dopamine levels in zebrafish lacking β-and γ^1-synucleins," *J. Biol. Chem.*, vol. 287, no. 5, pp. 2971–2983, Jan. 2012.

[20] J. N. Kay, K. C. Finger-Baier, T. Roeser, W. Staub, and H. Baier, "Retinal ganglion cell genesis requires Lakritz, a zebrafish atonal homolog," *Neuron*, vol. 30, no. 3, pp. 725–736, May 2001.

[21] S. Saleem and R. R. Kannan, "Zebrafish: An emerging real-time model system to study Alzheimer's disease and neurospecific drug discovery," *Cell Death Discovery*, vol. 4, no. 1, pp. 1–13, Dec. 2018.

[22] S. Krishna, K. Chatti, and R. R. Galigekere, "Automatic and robust estimation of heart rate in zebrafish larvae," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 3, pp. 1041–1052, Jul. 2018.

[23] J. Green *et al.*, "Automated high-throughput neurophenotyping of zebrafish social behavior," *J. Neurosci. Methods*, vol. 210, no. 2, pp. 266–271, Sep. 2012.

[24] A. Pérez-Escudero, J. Vicente-Page, R. C. Hinz, S. Arganda, and G. G. de Polavieja, "IdTracker: Tracking individuals in a group by automatic identification of unmarked animals," *Nature Methods*, vol. 11, no. 7, pp. 743–748, Jul. 2014.

[25] L. P. J. J. Noldus, A. J. Spink, and R. A. J. Tegelenbosch, "EthoVision: A versatile video tracking system for automation of behavioral experiments," *Behav. Res. Methods, Instrum., Comput.*, vol. 33, no. 3, pp. 398–414, Aug. 2001.

[26] C. Nadeau, H. Ren, A. Krupa, and P. Dupont, "Intensity-based visual servoing for instrument and tissue tracking in 3D ultrasound volumes," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 1, pp. 367–371, Jan. 2015.

[27] J. Wang, S. Song, H. Ren, C. M. Lim, and M. Q.-H. Meng, "Surgical instrument tracking by multiple monocular modules and a sensor fusion approach," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 2, pp. 629–639, Apr. 2019.

[28] H. Ren, W. Liu, and A. Lim, "Marker-based surgical instrument tracking using dual kinect sensors," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 3, pp. 921–924, Jul. 2014.

[29] J. Delcourt, M. Ylieff, V. Bolliet, P. Poncin, and A. Bardonnet, "Video tracking in the extreme: A new possibility for tracking nocturnal underwater transparent animals with fluorescent elastomer tags," *Behav. Res. Methods*, vol. 43, no. 2, pp. 590–600, Jun. 2011.

[30] A. Rodriguez, H. Zhang, J. Klaminder, T. Brodin, and M. Andersson, "ToxId: An efficient algorithm to solve occlusions when tracking multiple animals," *Sci. Rep.*, vol. 7, no. 1, pp. 1–8, Dec. 2017.

[31] T. Li, H. Chen, S. Sun, and J. M. Corchado, "Joint smoothing and tracking based on continuous-time target trajectory function fitting," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 3, pp. 1476–1483, Jul. 2019.

[32] C. Dai *et al.*, "Automated non-invasive measurement of single sperm's motility and morphology," *IEEE Trans. Med. Imag.*, vol. 37, no. 10, pp. 2257–2265, May 2018.

[33] S. H. Wang, X. E. Cheng, Z.-M. Qian, Y. Liu, and Y. Q. Chen, "Automated planar tracking the waving bodies of multiple zebrafish swimming in shallow water," *PLoS ONE*, vol. 11, no. 4, Apr. 2016, Art. no. e0154714.

[34] Y. Zhou, R. T. Cattley, C. L. Cario, Q. Bai, and E. A. Burton, "Quantification of larval zebrafish motor function in multiwell plates using open-source MATLAB applications," *Nature Protocols*, vol. 9, no. 7, pp. 1533–1548, Jul. 2014.

[35] F. Romero-Ferrero *et al.*, "Idtracker. Ai: Tracking all individuals in small or large collectives of unmarked animals," *Nature Methods*, vol. 16, no. 2, pp. 179–182, 2019.

[36] O. Mirat, J. R. Sternberg, K. E. Severi, and C. Wyart, "ZebraZoom: An automated program for high-throughput behavioral analysis and categorization," *Frontiers Neural Circuits*, vol. 7, pp. 1–12, 2013.

[37] X. Wang, E. Cheng, I. S. Burnett, Y. Huang, and D. Wlodkowic, "Automatic multiple zebrafish larvae tracking in unconstrained microscopic video conditions," *Sci. Rep.*, vol. 7, no. 1, pp. 1–8, Dec. 2017.

[38] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*. [Online]. Available: http://arxiv.org/abs/2004.10934

[39] Z. Zivkovic and F. van der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern Recognit. Lett.*, vol. 27, no. 7, pp. 773–780, May 2006.

[40] J. Delcourt, M. Denoël, M. Ylieff, and P. Poncin, "Video multitracking of fish behaviour: A synthesis and future perspectives," *Fish Fisheries*, vol. 14, no. 2, pp. 186–204, Jun. 2013.

[41] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Commun. ACM*, vol. 27, no. 3, pp. 236–239, Mar. 1984.

[42] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.

[43] K. Wagstaff *et al.*, "Constrained k-means clustering with background knowledge," in *Proc. ICML*, vol. 1, 2001, pp. 577–584.

[44] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation With Applications to Tracking and Navigation: Theory Algorithms and Software*. Hoboken, NJ, USA: Wiley, 2004.

[45] H. W. Kuhn, "The hungarian method for the assignment problem," *Nav. Res. Logistics Quart.*, vol. 2, nos. 1–2, pp. 83–97, Mar. 1955.

[46] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *EURASIP J. Image Video Process.*, vol. 2008, no. 1, pp. 1–10, 2008.

[47] R. Portugues and F. Engert, "The neural basis of visual behaviors in the larval zebrafish," *Current Opinion Neurobiol.*, vol. 19, no. 6, pp. 644–647, Dec. 2009.

[48] A. Weissbrod *et al.*, "Automated long-term tracking and social behavioural phenotyping of animal colonies within a semi-natural environment," *Nature Commun.*, vol. 4, no. 1, pp. 1–10, Oct. 2013.

[49] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.

Chunxiang Wang received the B.E. degree in automation from the Harbin Institute of Technology, Harbin, China, in 2019, where he is currently pursuing the master's degree in control science and engineering.

His research interests include image processing, object tracking, Kalman Filter and its improvements, and robotic micromanipulation with its biomedical applications.

Mingsi Tong (Member, IEEE) received the Ph.D. degree in mechatronic engineering from the Harbin Institute of Technology, Harbin, China, in 2016.

During his graduate period, he also concurrently studied at the National Institute of Standards and Technology, Washington, DC, USA, as a Guest Researcher. He is currently an Assistant Professor with the School of Mechatronics Engineering, Harbin Institute of Technology. His research interests include forensic science, pattern recognition, and surface metrology.

Liqun Zhao is currently pursuing the bachelor's degree in automation with the School of Astronautics, Harbin Institute of Technology, Harbin, China.

During his undergraduate period, he studied at The University of Adelaide, Adelaide, SA, Australia, as an international exchange student. His research interests include soft manipulators, medical imaging, decision-making, and trajectory planning of unmanned vehicles.

Xinghu Yu was born in Yantai, China, in 1988. He received the M.M. degree in osteopathic medicine from Jinzhou Medical University, Jinzhou, China, in 2016, and the Ph.D. degree in control science and engineering from the Harbin Institute of Technology, Harbin, China, in 2020.

He is currently the Chief Executive Officer with Ningbo Institute of Intelligent Equipment Technology Corporation, Ningbo, China. His research interests include switched systems, intelligent control, and biomedical image processing.

Songlin Zhuang (Member, IEEE) received the B.E. degree in automation and the Ph.D. degree in control science and engineering from the Harbin Institute of Technology, Harbin, China, in 2014 and 2019, respectively.

Since October 2019, he has been a Post-Doctoral Fellow with the Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, ON, Canada. His research interests include switched systems, model predictive control, and robotic micromanipulation with its biomedical applications.

Huijun Gao (Fellow, IEEE) received the Ph.D. degree in control science and engineering from the Harbin Institute of Technology, Harbin, China, in 2005.

From 2005 to 2007, he carried out his post-doctoral research at the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. Since 2004, he has been with the Harbin Institute of Technology, where he is currently a Full Professor and the Director of the Research Institute of Intelligent Control and Systems and the Interdisciplinary Research Center. His research interests include intelligent and robust control, robotics, mechatronics, and their engineering applications.

Dr. Gao is the Vice President of the IEEE Industrial Electronics Society (IES) and the Council Member of IFAC. He serves as the Coeditor-in-Chief for the IEEE Transactions on Industrial Electronics, a Senior Editor for the IEEE/ASME Transactions on Mechatronics, and an Associate Editor for *Automatica*, the IEEE Transactions on Cybernetics, and the IEEE Transactions on Industrial Informatics. He is a Distinguished Lecturer of the IEEE Systems, Man and Cybernetics Society.