3D-ZeF: A 3D Zebrafish Tracking Benchmark Dataset

Malte Pedersen*, Joakim Bruslund Haurum*, Stefan Hein Bengtson, Thomas B. Moeslund
Visual Analysis of People (VAP) Laboratory, Aalborg University, Denmark
mape@create.aau.dk, joha@create.aau.dk, shbe@create.aau.dk, tbm@create.aau.dk

Abstract

In this work we present a novel publicly available stereo based 3D RGB dataset for multi-object zebrafish tracking, called 3D-ZeF. Zebrafish is an increasingly popular model organism used for studying neurological disorders, drug addiction, and more. Behavioral analysis is often a critical part of such research. However, visual similarity, occlusion, and erratic movement of the zebrafish makes robust 3D tracking a challenging and unsolved problem.

The proposed dataset consists of eight sequences with a duration between 15-120 seconds and 1-10 free moving zebrafish. The videos have been annotated with a total of 86,400 points and bounding boxes. Furthermore, we present a complexity score and a novel open-source modular baseline system for 3D tracking of zebrafish. The performance of the system is measured with respect to two detectors: a naive approach and a Faster R-CNN based fish head detector. The system reaches a MOTA of up to 77.6%. Links to the code and dataset is available at the project page https://vap.aau.dk/3d-zef

1. Introduction

Over the past decades, the use of zebrafish (Danio rerio) as an animal model has increased significantly due to its applicability within large-scale genetic screening [1,2]. The zebrafish has been used as a model for studying human neurological disorders, drug addiction, social anxiety disorders, and more [3,4,5,6,7,8]. Locomotion and behavioral analysis are often critical parts of neuroscientific and biological research, which have traditionally been conducted manually [9,10,11]. However, manual inspection is subjective and limited to small-scale experiments. Therefore, tracking systems are getting increasingly popular due to their efficiency and objectivity. The majority of the solutions has been developed for terrestrial animals or fish in shallow water, and most studies have been based on 2D observations in scientific [12,13,14,15,16,17,18] and commercial systems [19,20,21,22]. However, observations in a single plane cannot capture all the relevant phenotypes of fish [23,24,25]. Estimating the 3D trajectories of multiple zebrafish accurately is difficult due to their erratic movement, visual similarity, and social behavior [26]. This may be one of the reasons why no commercial solution has been developed yet. Only few groups in the scientific community have addressed the problem, focusing mainly on stereo vision [27,28,29,30,31] and monocular stereo using mirrors [32,33]. However, no labeled datasets have been made publicly available within the field, which makes a fair comparison between the applied methods difficult. This ultimately hinders significant developments in the field as we have seen in other computer vision fields with common datasets. Therefore, our contributions are

- a publicly available RGB 3D video dataset of zebrafish with 86,400 bounding box and point annotations.
- an open-source modular baseline system.

A large part of 3D multi-object tracking methods are developed for LiDAR-based traffic datasets [34,35,36,37,38] or RGB-D tracking [39,40]. However, to the best of our knowledge, there exists no publicly available annotated RGB stereo dataset with erratic moving and similarly looking subjects like the one we propose.
2. Related Work

Multi-Object Tracking (MOT). Reliably tracking multiple objects is widely regarded as incredibly difficult. The interest in solving MOT has been steadily increasing since 2015 with the release of the MOT [41, 42, 43], UA-DETRAC [44, 45], and KITTI [44, 45] challenges. Within the MOT challenges, the current focus is on either aiming to solve the association problem using deep learning [46], using techniques such as intersection-over-union based tracking [47], or disregarding tracking-specific models and utilizing the improvements within object detections [48].

Zebrafish Tracking. Vision-based tracking systems developed for studying animal behavior have traditionally been based on 2D [18, 49, 50, 51, 52, 53, 54] due to simplicity and because the movement of most terrestrial animals can be approximated to a single plane. The majority of research in zebrafish tracking has followed this path by only allowing the fish to move in shallow water and assuming that motion happens in a 2D plane.

A 2D animal tracker, called idTracker presented by Perez-Escudero et al. in 2014 [49], uses thresholding to segment blobs and is able to distinguish between individual zebrafish based on intensity and contrast maps. In 2019, Romero-Ferrero et al. presented an updated version of idTracker, called idtracker.ai [18], which is the current state-of-the-art 2D tracker system based on convolutional neural networks (CNN) for handling occlusions and identifying individuals. The subjects are observed with a camera positioned above a tank with a water depth of 2.5 cm and the distance between camera and subjects is, therefore, approximately the same at all times. As stated by the authors, this simplifies the task compared to a real 3D tracking scenario.

However, as most aquatic species move in three dimensions, trajectories in 3D are required to thoroughly describe their behavior [55, 56]. The most frequently used acquisition method when dealing with studies of animal behavior in 3D is stereo vision [28, 30, 31, 56, 57, 58, 59, 60, 61]. 3D tracking of zebrafish has been focused mainly on single subjects or small groups, as occlusion is a big hindrance for maintaining correct IDs due to their shoaling behavior [26]. Furthermore, the visual appearance of the fish can change dramatically depending on the position and posture, which makes re-identification more complex compared to 2D.

The Track3D module from the commercial EthoVision XT [19] is popular for tracking zebrafish in 3D, but is limited to a single individual [56, 61]. An early semi-automatic 3D tracking system was developed by Viscido et al. [58] to investigate the relationship between individual members of fish schools. Initial 2D tracks were generated by a nearest neighbor algorithm followed by a step allowing the user to adjust and correct the proposed 2D trajectories, and subsequently triangulated to reconstruct the 3D trajectories.

Qian et al. have worked extensively with tracking of zebrafish and have developed a 2D tracking system with a top-view camera using an augmented fast marching method (AFMM) [62] and the determinant of the Hessian [15]. This was expanded to 3D tracking by extending the setup with a side-view camera. AFMM was utilized to generate a feature point based fish representation in each view followed by 2D tracklet construction based on motion constraints. 3D tracks were then constructed by associating the 2D tracklets with side-view detections using epipolar and motion consistency constraints [29]. Liu et al. [63] extended this method to better handle occlusions based on a set of heuristic methods and the epipolar constraint. A third camera was added in [31], and the feature point representation method was extended.

Cheng et al. [28] utilized a similar three-camera setup, applying an iterative unsupervised learning method to train a CNN-based classifier to distinguish between the individual fish from a camera placed above the water tank. The classifier was trained on the head region of the fish during periods when all fish were visible at the same time. By iteratively retraining the classifier, they were able to generate 2D tracks from the top-view and reconstruct the 3D tracklets based on detections from the two other side-view cameras under epipolar and motion constraints.

Wang et al. [30] also utilized a three-camera setup, using a Gaussian Mixture Model, a Gabor filter and an SVM-based method to detect the fish heads in the top- and side-
3. Proposed Dataset

The proposed 3D zebrafish dataset, 3D-ZeF, has been recorded from a top- and front-view perspective. This approach was taken to minimize events of total occlusion typical for side-by-side binocular setups. An example of the visual variation between the views is shown in Figure 2 together with an illustration of the experimental setup.

3.1. Experimental Setup

The setup used to record the proposed dataset was built entirely from off-the-shelf hardware, whereas previous methods have used specialized camera equipment. An illustration of the setup is shown in Figure 2. The two light panels are IKEA FLOALT of size 30 × 30 cm with a luminous flux of 670 lumen and a color temperature of 4000K. The test tank is a standard glass aquarium of size 30 × 30 × 30 cm with a water depth of 15 cm. The top and front cameras are GoPro Hero 5 and GoPro Hero 7, respectively. All the videos are recorded with a resolution of 2704 × 1520, 60 FPS, 1/60 s shutter speed, 400 ISO, and a linear field of view. However, the fish tank does not take up the entire image, therefore, the effective region of interest is approximated as

\[
\text{Length} = 120 \text{s}, 15 \text{s}, 30 \text{s}, 15 \text{s} \\
\text{Frames} = 14,400, 1,800, 3,600, 1,800 \\
\text{BBs} = 28,800, 9,000, 7,200, 9,000 \\
\text{Points} = 28,800, 9,000, 7,200, 9,000
\]

Table 1: Overview of the proposed dataset. OC, OL, TBO, and IBO are listed for the top- and front-view, respectively, and the number of fish is denoted in the sequence name. OC: average amount of occlusions per second, OL: average occlusion length in seconds, TBO: average amount of seconds between occlusions, IBO: intersection between occlusions, \( \Psi \): complexity measure based on OC, OL, TBO and IBO (see Equation (2)).
involved bounding boxes. During occlusions, the bounding box was fitted to the visible parts of the fish and not where it was expected to be due to the extreme flexibility of the zebrafish. The pair-wise point annotations from the two views were triangulated into 3D positions using the method proposed by Pedersen et al. [67]. The fish head was approximated during occlusions to ensure continuous 3D tracks.

It should be noted that the data was recorded in RGB. Zebrafish can change their body pigmentation based on their environment, stress level, and more [23]. The changes in coloration can be important in behavioral studies and may even be valuable in solving the 3D tracking problem.

3.3. Dataset Complexity

Intuitively, a higher number of fish creates a more difficult tracking problem. However, this is only true to some extent as the main complexity factor is the number and level of occlusions, which depends on a combination of the social activity and amount of space rather than the number of individuals. Therefore, we have defined a range of metrics based on occlusion events to describe the complexity of the proposed sequences. An occlusion event is defined by a set of consecutive frames, where a fish is part of an occlusion. The events are measured from the perspective of the fish; if two fish are part of an occlusion it counts as two events.

The number of occlusion events indicates how often a fish is part of an occlusion, but, few long occlusions can be just as problematic as many short. The length of the occlusions and time between them are, therefore, important to keep in mind when evaluating the complexity of a recording. Due to our definition of occlusion events there are cases where fish are part of occlusions with only minor parts of their bodies. Therefore, the intersection between occlusions is measured as an indication of the general intersection level. The metrics that we provide as basis for the complexity level of our recordings are defined here:

**Occlusion Count (OC):** the average number of occlusion events per second.

**Occlusion Length (OL):** the average time in seconds of all occlusion events.

**Time Between Occlusions (TBO):** the average time in seconds between occlusion events.

**Intersection Between Occlusions (IBO):** a measure of how large a part of the fish that is part of an occlusion event. The intersection in a frame, $f$, for fish $i$ is given by

$$ IBO_{i,f} = \frac{1}{n_{occ}} \sum_{j=1}^{n_{occ}} bb_i \cap bb_j, \quad \text{for} \ j \neq i, \quad (1) $$

where $n_{occ}$ is the number of fish in an occlusion event, and $bb_j$ is the set of pixel coordinates in the bounding box of fish $j$. IBO is measured across all bounding boxes with an occlusion tag in a given frame, even for subjects that are not part of the same occlusion. Two examples are presented in Figure 3 where the $IBO_{i,f}$ is calculated from the perspective of the targets enclosed in yellow. The blue area in the second example, represents the intersection with a subject that is not part of the same occlusion as the target. Additionally, the annotated bounding boxes enclose only the visible parts of the subjects. Thus, the actual intersection between the subjects can be higher if a large part of a fish is hidden. Nonetheless, the assumption is that a high IBO is an expression of heavy occlusion and vice versa. The IBO measure presented in Table 1 is an average between all fish in all frames. A single complexity measure is calculated per sequence, by combining the four proposed metrics by

$$ \Psi = \frac{1}{n} \sum_v \left( \frac{\text{OC}_v \ \text{OL}_v \ \text{IBO}_v}{\text{TBO}_v} \right), \quad (2) $$

where $n$ is the number of camera views and subscript T and F denote the top- and front-view, respectively. If a recording has no occlusions the complexity measure, $\Psi$, is zero; otherwise, the measure is in the interval $[0, \infty]$, where a larger value indicates a higher complexity.

4. Method

The pipeline of the proposed 3D tracker follows a mod-}

ular tracking-reconstruction approach, where subjects are detected and tracked in each view before being triangulated and associated across views. This allows us to use the temporal information of the tracklets in the two views in the 3D association step in opposition to a reconstruction-tracking approach, where detections are triangulated before tracks are generated.

4.1. Object Detection in 2D

A consistent 2D point is needed in each view in order to create 3D trajectories. As the head is the only rigid part of
the body the tracking point is chosen to be located between the eyes of the fish. We present two simple methods to find the head-point of the fish: a naive approach, that does not require training, and a CNN based approach.

**Naive**: A background image, $b_g$, is initially estimated for each view by taking the median of $N_{bg}$ images sampled uniformly across the videos. Subsequently, the background is subtracted by calculating the absolute difference image, $f_g = |im - bg|$. To locate the head of a fish in the top-view, the $f_g$ is binarized using the internodes bimodal threshold algorithm [68]. The skeletonization approach of Zhang and Suen [69] is applied, and the endpoints are analyzed to locate the head of the fish. In the front-view the $f_g$ is binarized through the use of a histogram entropy thresholding method because the appearance of the fish cannot be approximated as bimodal. The head point is estimated as being either the center of the blob or one of the middle edge points of the blob along the minor axis of the detected bounding box. All three points are evaluated during the 3D reconstruction step, and the two points with the highest reprojection errors are discarded.

**FRCNN-H**: A Faster R-CNN [70] model has been trained for each view. The bounding boxes have been extracted from all the head-point annotations in the training sequences in order to train a head-detector model for each view. The bounding boxes have static diameters of 25 and 50 pixels for the top- and front-view, respectively. The head-points are determined as the center of the detected bounding boxes which have a minimum confidence of $c$. See the supplementary material for more detailed information on the detectors.

### 4.2. 2D Tracklet Construction

As zebrafish move erratically, it is difficult to set up a stable motion model. Therefore, we use a naive tracking-by-detection approach. The tracking is done by constructing a distance matrix between the detections in a frame and the last detections of current tracklets. The matrix is solved as a global optimization problem using the Hungarian algorithm [71]. Tracklets are deliberately constructed in a conservative manner, where robustness is encouraged above length. A new detection is only assigned to a tracklet located within a minimum distance, denoted $\delta_T$ and $\delta_F$, for the top and front view respectively. If a tracklet has not been assigned a detection within a given amount of time, $\tau_k$, the tracklet is terminated.

The $\ell_2$ distance between the head detections is used in both views for the FRCNN-H method. However, the Mahalanobis distance between the center-of-mass is used for the front-view in the Naive method. This is due to the elliptical form of the zebrafish body, which can be utilized by setting the covariance matrix of the blob as the Mahalanobis matrix; as the fish is more likely to move along the major axis than along the minor axis.

### 4.3. 2D Tracklet Association Between Views

The 2D tracklets from each view are associated into 3D tracklets through a graph-based approach. All 2D tracklets with less than a given number of detections, $\alpha$, are removed in order to filter out noisy tracklets. The 3D calibration and triangulation method from Pedersen et al. [67] is used.

#### 4.3.1 Graph Construction

A directed acyclic graph (DAG) is constructed. Every node represents a 3D tracklet and consists of two 2D tracklets; one from each camera view. Each edge associates nodes, where the 3D tracklet is based on the same 2D tracklet from one of the views.

**Create nodes**: The graph nodes are constructed by processing each top-view tracklet and identifying all temporally intersecting front-view tracklets as given by

$$I = F_T \cap F_F,$$

where $F_T$ and $F_F$ are the set of frames with detections in the top- and front-view tracklets, respectively, and $I$ is the set of frames with detections in both views. If $I = \emptyset$, the node is not created.

An example is presented in Figure 4, where both the blue and red tracklets in the top-view intersects with the three tracklets in the front-view. The outer and inner circles of the six nodes represent the top- and front-view tracklets, respectively. The number inside the nodes indicates the node weight, which is calculated as follows.

For each intersecting frame in $I$, denoted $f$, the 2D tracklets are triangulated. This results in a 3D point of the zebrafish head, $p_f$, with a reprojection error, $x_f$. For the Naive method where the head is not directly detected in the front-view, the top-view 2D point is triangulated with the three estimated points to find the match resulting in the smallest reprojection error. Therefore, $p_f$ represents the point with the smallest reprojection error. To penalize large reprojection errors, the distance between the head detections is used in both views for the FRCNN-H method.

![Figure 4: The colored lines represent 2D tracklets in each view, the node pairs are represented by the double-colored circles, and the edges of the DAG are shown by the arrows. The numbers represent example node and edge weights.](image)

---

[Image 316x637 to 538x720]
errors, the complimentary probability from the exponential cumulative distribution function (CDF), $\Phi$, is utilized. The exponential CDF is chosen as it approximately models the reprojection error of the ground truth training data. The set of weights for all valid 3D points, $V$, can be described by the following set-builder notation

$$V = \{1 - \Phi(x_f | \lambda_{err}) \mid f \in I \land A(p_f)\}, \quad (4)$$

where $\lambda_{err}$ is the reciprocal of the mean of the training data reprojection error, and $A$ states whether $p_f$ is within the water tank. The per-frame weights in $V$ are combined into a single weight, $W$, for the entire node by

$$W = \text{median}(V) \frac{|V|}{|F_T \cup F_{T'}|}, \quad (5)$$

and the node is added to the DAG given that $W \neq 0$. This weighting scheme considers both the reprojection error and the ratio of frames with valid 3D points compared to the set of all frames $I$. The median function is used instead of the mean function in order to counteract that a few extreme outliers skew the weight.

**Connect nodes:** The nodes in the DAG should be connected to all other nodes building on one of the same 2D tracklets, as long as the 2D tracklets in the other view do not overlap temporally, as illustrated in Figure 4. This is done by constructing the set of node pairs, $P$, from the set of nodes in the DAG, $N$. Each element of $N$, denoted $n$, consists of the 2D tracklets, $t_F$ and $t_{T'}$, the 3D tracklet, $t$, and the node weight, $W$. Nodes $n_i$ and $n_j$ are considered a pair if $t_{i,F} = t_{j,T}$ or $t_{i,T'} = t_{j,F}$, if the 2D tracklets in the other view do not temporally overlap, and if $t_i$ starts earlier in time than $t_j$. This is necessary in order to avoid assigning multiple detections to the same frame.

This can be represented by the set-builder notation

$$P = \{(n_i, n_j) \mid n_i, n_j \in N \land O(n_i, n_j) \land T(n_i, n_j)\}, \quad (6)$$

where $O$ assesses whether $t_i$ starts before $t_j$, and $T$ ensures that the 2D tracklets in $n_i$ and $n_j$ do not temporally overlap, where $n = \{t_T, t_F, t, W\}$.

For each node pair in $P$, the weight, $E$, of the directed edge from $n_i$ to $n_j$ is based on:

- $s$, the speed of the fish as it moves between the last detection in $t_i$ and the first detection in $t_j$,
- $t_d$, the temporal difference between $t_i$ and $t_j$,
- $W_i$ and $W_j$, the weights of the nodes.

The edge weight is calculated as the complimentary probability of the CDF of the exponential distribution, $\Phi$. The exponential distribution is chosen as it approximately models that of the speed of the zebrafish. $E$ is calculated by

$$E = (1 - \Phi(s | \lambda_s))e^{-\frac{t_d}{\tau_p}} (W_i + W_j), \quad (7)$$

where $\tau_p$ is an empirically chosen value, and $\lambda_s$ is the reciprocal of the sum of the mean and standard deviation of the measured speed in the training data. In case a node is not present in any node pairs, the node will be assigned to the DAG, but it will have no edges. The DAG is therefore a disconnected graph.

**4.3.2 Graph Evaluation**

The final 3D tracklets are extracted from the constructed DAG; this is done by recursively finding the longest path in the graph and storing the set of nodes as a single 3D tracklet. The longest path is the path throughout the DAG, which gives the highest value when summing all nodes and edge weights in the path, see Figure 5. After extraction of a path, the used nodes, and all other nodes using the same 2D tracklets, are removed from the DAG. This process is repeated until the DAG is empty. In case a 2D tracklet in the 3D tracklet is missing a detection, the 3D position cannot be assigned, but the known information of the 2D tracklet is kept. For the Naive method, the head position of the front-view 2D tracklet is determined by assigning the estimated point, which minimizes the $\ell_2$ distance to the head positions in the consecutive frame.

**4.4. 3D Tracklet Association**

The final 3D tracks are constructed from the 3D tracklets in a greedy manner. A set of tracklets equal to the amount of fish present, $N_{\text{fish}}$, is used as initial main tracklets. The remaining tracklets, denoted gallery tracklets, are assigned one by one to a single main tracklet, until no more tracklets can be assigned.

**4.4.1 Initial Tracklet Selection**

The set of $N_{\text{fish}}$ in the main tracks is selected by finding the stable tracklets that are temporally concurrent in time and span long time intervals. For each tracklet, the set of other temporally concurrent tracklets is considered. In this set, all possible combinations of size $N_{\text{fish}}$ are investigated. If
all tracklets in the set overlap temporally, the set is saved as a valid tracklet set. The valid tracklet set with the highest median temporal overlap is used to construct $N_{fish}$ full 3D tracks. This is done by using the greedy association scheme described in the following sections. No 3D tracks are created if no valid combination of size $N_{fish}$ is identified.

### 4.4.2 Greedy Association

A greedy association algorithm is used when each gallery tracklet is associated with a single main tracklet. The greedy part of the algorithm concerns the way that gallery tracklets are chosen; all gallery tracks are ranked in ascending order by the shortest temporal distance to any main tracklet. If the gallery tracklet overlaps temporally with all main tracklets, it is relegated to the end of the list. When the gallery tracklet has been associated with a main track, the remaining gallery tracks are re-ranked, and the process repeated. In this way, the main tracklets are “grown” into full tracks. The gallery tracklet assignment is based on minimizing the cost of assignment. The cost is based on a set of distance measures, which are determined from two cases.

In the first case at least one main tracklet does not temporally overlap with the gallery tracklet. In this case, the association process is based on the spatio-temporal distances between the gallery tracklet and main tracklets. All temporally overlapping main tracklets are not considered.

In the second case the gallery tracklet overlaps temporally with all main tracklets. As the spatio-temporal distances between the main and gallery tracklet is no longer measurable, a different set of distance values are used: The internal spatio-temporal distances, the amount of intersecting frames, i.e., frames with a detection in both the main and gallery tracklets, and the ratio of intersecting frames compared to the total amount of detections in the gallery tracklet. The internal spatio-temporal distances are determined through the construction of a DAG, where each node is a detection in a frame, and the edge weights are the spatial distances between the temporally previous nodes. The final path is the one minimizing the spatial distance traveled. An example of a graph is shown in Figure 6. The distances are calculated as the mean of the values when the graph switches from a detection in the gallery tracklet to the main tracklet and vice versa.

**Association:** The distance measures are consolidated into a single assignment decision through a global cost scheme. Each distance value is normalized across valid main tracklets into the range $[0; 1]$ and sum to 1. The final cost of assigning the gallery tracklet to a main tracklet, is obtained by calculating the mean of the normalized distance values. The gallery tracklet is associated with the main tracklet with the smallest cost, unless all main tracklet costs are located within a small margin, $\beta$, of each other, in which case the gallery tracklet is discarded. $\beta$ directly enforces a margin of confidence in the assignment, in order to not assign a gallery tracklet based on inconclusive cost values.

## 5. Evaluation

The metrics used in the MOT challenges [41, 42, 43] and the Mean Time Between Failures (MTBF) proposed by Carr and Collins [72] are utilized to measure the performance of the system on the proposed dataset. The MOT challenge metrics consist of the CLEAR MOT metrics [73], the mostly tracked/lost metrics [74], and the identification-based metrics [75].

The final 3D tracks are evaluated based on a subset of the MOT challenge metrics and the monotonic MTBF metric. The primary metric used is the multiple object tracking (MOTA) metric. The detected and ground truth tracklets are compared using the detected and annotated head points. A detection is only associated with a ground truth tracklet if it is within a distance of 0.5 cm. The performance of the system is evaluated with two different detection modules: Naive, and FRCNN-H. The results are compared with a hypothetical tracker, called Oracle, which tracks perfectly at all times except during occlusions. This provides an upper bound on the performance if occlusions are not handled in any way. The full set of metrics, system parameters, and results can be found in the supplementary material.

Results for all sequences compared to data complexity is shown in Figure 7, and metrics for the test sequences are shown in Table 2. It is clear that the FRCNN-H outperforms the Naive method on the training and validation splits; it even outperforms the Oracle tracker in three out of four cases. This is likely due to the method being able to detect some of the fish heads during occlusions. However, the superior performance is only seen on the two splits where the fish are from the same cohort. On the test set the FRCNN-H fails to generalize, whereas the Naive method still manages to track the fish.
It should be noted that the poor performance of the Naive method on Tst1, is suspected to be due many short tracks from erratic movement, which the pipeline with the used parameter settings does not handle well.

5.1. Comparison with Other Methods

It has not been possible to make a fair comparison with the other 3D zebrafish tracking methods mentioned in Section 2. Previous systems have been analyzed in terms of ID swaps, fragments, precision, and recall for the generated 2D and 3D tracks. However, there is no exact description of how these metrics are calculated. The evaluation protocol is further limited by not including a statement on the maximum allowed distance between estimated and ground truth tracks leading to uncertainty on the accuracy of the metrics.

Furthermore, the evaluated sequences are not described in terms of complexity, even though occlusion is repeatedly stated as a major hindrance in 3D zebrafish tracking. The only common complexity indication of the datasets is the number of fish, even though it is not representative. An example of this is the tracking demo video of Qian et al. [62] with ten fish and only four occlusion events during 15 seconds. Wang et al. [30] describes their dataset on basis of an occlusion probability but do not explain how it is measured.

There are currently no publicly available annotated data and the previous systems are evaluated on seemingly simplified cases of the problem. Furthermore, the used evaluation protocols are lacking details in such a manner that it is not possible to determine under which conditions the metrics have been calculated. This, along with inaccessible codebases, severely limits the reproducibility of the results, and makes it impossible to ensure identical evaluation procedures. Therefore, it simply does not make sense to compare the proposed system to the other methods under the current circumstances.

6. Conclusion

Zebrafish is an increasingly popular animal model and behavioral analysis plays a major role in neuroscientific and biological research. However, it is tedious and subjective to manually describe the complex 3D motion of zebrafish. Therefore, 3D zebrafish tracking systems are critically needed to conduct accurate experiments on a grand scale. The significant development experienced in other fields of MOT has not yet translated to 3D zebrafish tracking. The main reason being that no dataset has been made publicly available with ground truth annotations. Therefore, we present the first publicly available RGB 3D zebrafish tracking dataset called 3D-ZeF.

3D-ZeF consists of eight stereo sequences with highly social and similarly looking subjects demonstrating complex and erratic motion patterns in three dimensions that are not seen in common MOT challenges. A complexity measure based on the level of occlusions has been provided for each sequence to make them comparable to future related datasets. The proposed dataset is annotated with 86,400 bounding boxes and points; the latter used for estimating ground truth 3D tracks based on the head position of the fish. Different cohorts of zebrafish are used in the training, validation, and test splits to avoid data leakage; a problem that has never been addressed within the field.

The proposed Naive method scores a MOTA between 25% and 80% across the entire dataset, which correlates well with the complexity measure of the recordings. The open-source modular based system provides a baseline and stepping stone for further development within the field of 3D zebrafish tracking and understanding.
References

[1] J. S. Eisen, “Zebrafish make a big splash,” Cell, vol. 87, pp. 969–977, Dec. 1996.

[2] P. Haffter, M. Granato, M. Brand, M. Mullins, M. Hammerschmidt, D. Kane, J. Odenthal, F. van Eeden, Y. Jiang, C. Heisenberg, R. Kelsch, M. Furutani-Seiki, E. Vogelsang, D. Beuchle, U. Schach, C. Fabian, and C. Nusslein-Volhard, “The identification of genes with unique and essential functions in the development of the zebrafish, Danio rerio,” Development, vol. 123, no. 1, pp. 1–36, 1996.

[3] A. D. Collier, K. M. Khan, E. M. Caramillo, R. S. Mohn, and D. J. Echevarria, “Zebrafish and conditioned place preference: A translational model of drug reward,” Progress in Neuro-Psychopharmacology and Biological Psychiatry, vol. 55, pp. 16–25, Dec. 2014.

[4] C.-Y. Lin, C.-Y. Chiang, and H.-J. Tsai, “Zebrafish and Medaka: new model organisms for modern biomedical research,” Journal of Biomedical Science, vol. 23, Jan. 2016.

[5] M. C. Soares, S. C. Cardoso, T. d. S. Carvalho, and C. Maximino, “Using model fish to study the biological mechanisms of cooperative behaviour: A future for translational research concerning social anxiety disorders?,” Progress in Neuro-Psychopharmacology and Biological Psychiatry, vol. 82, pp. 205–215, Mar. 2018.

[6] A. V. Kaluﬀ, A. M. Stewart, and R. Gerlai, “Zebrafish as an emerging model for studying complex brain disorders,” Trends in Pharmacological Sciences, vol. 35, pp. 63–75, Feb. 2014.

[7] D. A. Meshalkina, M. N. Kizlyk, E. V. Kysil, A. D. Collier, D. J. Echevarria, M. S. Abreu, L. J. G. Barcellos, C. Song, J. E. Warnick, E. J. Kyzar, and A. V. Kaluﬀ, “Zebrafish models of autism spectrum disorder,” Experimental Neurology, vol. 299, pp. 207–216, Jan. 2018.

[8] K. M. Khan, A. D. Collier, D. A. Meshalkina, E. V. Kysil, S. L. Khatsko, T. Kolosnikova, Y. Y. Morzerhin, J. E. Warnick, A. V. Kaluﬀ, and D. J. Echevarria, “Zebrafish models in neuropsychopharmacology and CNS drug discovery,” British Journal of Pharmacology, vol. 174, no. 13, pp. 1925–1944, 2017.

[9] L. Li and J. E. Dowling, “A dominant form of inherited retinal degeneration caused by a non-photoreceptor cell-speciﬁc mutation,” Proceedings of the National Academy of Sciences, vol. 94, pp. 11645–11650, Oct. 1997.

[10] U. K. Muller, “Swimming of larval zebrafish: ontogeny of body waves and implications for locomotory development,” Journal of Experimental Biology, vol. 207, pp. 853–868, Feb. 2004.

[11] M. B. McElligott and D. M. O’Malley, “Prey tracking by larval zebrafish: Axial kinematics and visual control,” Brain, Behavior and Evolution, vol. 66, no. 3, pp. 177–196, 2005.

[12] E. Fontaine, D. Lentink, S. Kranebarg, U. K. Muller, J. L. van Leeuwen, A. H. Barr, and J. W. Burdick, “Automated visual tracking for studying the ontogeny of zebrafish swimming,” Journal of Experimental Biology, vol. 211, no. 8, pp. 1305–1316, 2008.

[13] B. Risse, D. Berh, N. Otto, C. Klämbt, and X. Jiang, “FIMTrack: An open source tracking and locomotion analysis software for small animals,” PLOS Computational Biology, vol. 13, no. 5, pp. 1–15, 2017.

[14] S. Ohayon, O. Avni, A. L. Taylor, P. Perona, and S. E. R. Egnor, “Automated multi-day tracking of marked mice for the analysis of social behaviour,” Journal of Neuroscience Methods, vol. 219, no. 1, pp. 10 – 19, 2013.

[15] Z.-M. Qian, X. E. Cheng, and Y. Q. Chen, “Automatically detect and track multiple fish swimming in shallow water with frequent occlusion,” PLOS ONE, vol. 9, no. 9, pp. 1–12, 2014.

[16] G. d. O. Feijó, V. A. Sangalli, I. N. L. d. Silva, and M. S. Pinho, “An algorithm to track laboratory zebrafish shoals,” Computers in Biology and Medicine, vol. 96, pp. 79 – 90, 2018.

[17] J. E. Franco-Restrepo, D. A. Forero, and R. A. Vargas, “A review of freely available, open-source software for the automated analysis of the behavior of adult zebrafish,” Zebrafish, vol. 16, June 2019.

[18] F. Romero-Ferrero, M. G. Bergomi, R. C. Hinz, F. J. H. Heras, and G. G. d. Polavieja, “idtracker.ai: tracking all individuals in small or large collective of unmarked animals,” Nature Methods, vol. 16, pp. 179–182, Jan. 2019.

[19] L. P. J. J. Noldus, A. J. Spink, and R. A. J. Tegelenbosch, “EvoVision: A versatile video tracking system for automation of behavioral experiments,” Behavior Research Methods, Instruments, & Computers, vol. 33, pp. 398–414, Aug. 2001.

[20] Loligo Systems, “LoliTrack v.4.” https://www.loligosystems.com/lolitrack-v-4

[21] ViewPoint, “ZebraLab.” http://www.viewpoint.fr/en/p/software/zebralab

[22] TSE Systems, “VideoMot2 - versatile video tracking system.” https://www.tse-systems.com/product-details/videomot

[23] A. V. Kaluﬀ, M. Gebhardt, A. M. Stewart, J. M. Cachat, M. Brimmer, J. S. Chawla, C. Craddock, E. J. Kyzar, A. Roth, S. Landsman, S. Gaikwad, K. Robinson, E. Baatrup, K. Tierney, A. Shamchuk, W. Norton, N. Miller, T. Nicolson, O. Braubach, C. P. Gilman, J. Pittman, D. B. Rosenberger, R. Gerlai, D. Echevarria, E. Lamb, S. C. F. Neuhauss, W. Weng, L. Bally-Cuif, H. Schneider, and t. Z. Neurons, “Towards a comprehensive catalog of zebrafish behavior 1.0 and beyond,” Zebrafish, vol. 10, pp. 70–86, Mar. 2013.

[24] J. M. Cachat, P. R. Canavello, S. I. Elkayat, B. K. Bartels, P. C. Hart, M. F. Elegante, E. C. Beeson, A. L. Laffoon, W. A. Haymore, D. H. Tien, A. K. Tien, S. Mohnot, and A. V. Kaluﬀ, “Video-aided analysis of zebrafish locomotion and anxiety-related behavioral responses,” in Zebrafish Neurobehavioral Protocols (A. V. Kaluﬀ and J. M. Cachat, eds.), pp. 1–14, Totowa, NJ: Humana Press, 2011.
[51] H. J. Mönck, A. Jörg, T. v. Falkenhausen, J. Tanke, B. Wild, D. Dormagen, J. Piotrowski, C. Winklmayr, D. Bierbach, and T. Landgraf, “BioTracker: an open-source computer vision framework for visual animal tracking,” CoRR, vol. abs/1803.07985, 2018.

[52] A. Rodriguez, H. Zhang, J. Klaminder, T. Brodin, P. L. Andersson, and M. Andersson, “ToxTrac: a fast and robust software for tracking organisms,” Methods in Ecology and Evolution, vol. 9, pp. 460–464, Sept. 2017.

[53] A. M. T. Harmer and D. B. Thomas, “pathtrackr: An r package for video tracking and analysing animal movement,” Methods in Ecology and Evolution, May 2019.

[54] X. Liu, P. R. Zhu, Y. Liu, and J. W. Zhao, “Tracking full-body motion of multiple fish with midline subspace constrained multicue optimization,” Scientific Programming, vol. 2019, pp. 1–7, June 2019.

[55] L. Zhu and W. Weng, “Catadioptric stereo-vision system for the real-time monitoring of 3D behavior in aquatic animals,” Physiology & Behavior, vol. 91, no. 1, pp. 106 – 119, 2007.

[56] J. Cachat, A. Stewart, E. Utterback, P. Hart, S. Gaikwad, K. Wong, E. Kyzar, N. Wu, and A. V. Kaluuffe, “Three-dimensional neurophenotyping of adult zebrafish behavior,” PLOS ONE, vol. 6, no. 3, pp. 1–14, 2011.

[57] X. E. Cheng, S. H. Wang, and Y. Q. Chen, “3D tracking targets via kinematic model weighted particle filter,” in 2016 IEEE International Conference on Multimedia and Expo (ICME), pp. 1–6, July 2016.

[58] S. V. Viscido, J. K. Parrish, and D. Grünbaum, “Individual behavior and emergent properties of fish schools: a comparison of observation and theory,” Marine Ecology Progress Series, vol. 273, pp. 239–249, 2004.

[59] A. D. Straw, K. Branson, T. R. Neumann, and M. H. Dickinson, “Multi-camera real-time three-dimensional tracking of multiple flying animals,” Journal of The Royal Society Interface, vol. 8, pp. 395–409, July 2010.

[60] K. Müller, J. Schlemper, L. Kuhnert, and K. D. Kuhnert, “Calibration and 3D ground truth data generation with orthogonal camera-setup and refraction compensation for aquaria in real-time,” in 2014 International Conference on Computer Vision Theory and Applications (VISAPP), vol. 3, pp. 626–634, Jan. 2014.

[61] A. M. Stewart, F. Grieco, R. A. J. Tegelenbosch, E. J. Kyzar, M. Nguyen, A. Kaluueva, C. Song, L. P. J. J. Noldus, and A. V. Kaluuffe, “A novel 3DM method of locomotor analysis in adult zebrafish: Implications for automated detection of CNS drug-evoked phenotypes,” Journal of Neuroscience Methods, vol. 255, pp. 66–74, Nov. 2015.

[62] Z.-M. Qian, S. H. Wang, X. E. Cheng, and Y. Q. Chen, “An effective and robust method for tracking multiple fish in video image based on fish head detection,” BMC Bioinformatics, vol. 17, p. 251, June 2016.

[63] X. Liu, Y. Yue, M. Shi, and Z.-M. Qian, “3-D video tracking of multiple fish in a water tank,” IEEE Access, vol. 7, pp. 145049–145059, 2019.

[64] S. H. Wang, X. E. Cheng, and Y. Q. Chen, “Tracking undulatory body motion of multiple fish based on midline dynamics modeling,” in 2016 IEEE International Conference on Multimedia and Expo (ICME), pp. 1–6, July 2016.

[65] S. H. Wang, J. Zhao, X. Liu, Z. Qian, Y. Liu, and Y. Q. Chen, “3D tracking swimming fish school with learned kinematic model using LSTM network,” in 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1068–1072, March 2017.

[66] S. H. Wang, J. W. Zhao, and Y. Q. Chen, “Robust tracking of fish schools using CNN for head identification,” Multimedia Tools and Applications, pp. 1–19, 2016.

[67] M. Pedersen, S. Hein Bengtson, R. Gade, N. Madsen, and T. B. Moeslund, “Camera calibration for underwater 3D reconstruction based on ray tracing using Snell’s law,” in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, June 2018.

[68] J. M. S. Prewitt and M. L. Mendelsohn, “The analysis of cell images,” Annals of the New York Academy of Sciences, vol. 128, no. 3, pp. 1035–1053, 1966.

[69] T. Zhang and C. Y. Suen, “A fast parallel algorithm for thinning digital patterns,” Communications of the ACM, vol. 27, no. 3, pp. 236–239, 1984.

[70] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in Advances in Neural Information Processing Systems, pp. 91–99, 2015.

[71] H. W. Kuhn, “The Hungarian method for the assignment problem,” Naval Research Logistics Quarterly, vol. 2, no. 1-2, pp. 83–97, 1955.

[72] P. Carr and R. T. Collins, “Assessing tracking performance in complex scenarios using mean time between failures,” in 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1–10, March 2016.

[73] K. Bernardin and R. Stiefelhagen, “Evaluating multiple object tracking performance: The CLEAR MOT metrics,” EURASIP Journal on Image and Video Processing, vol. 2008, p. 246309, May 2008.

[74] Bo Wu and R. Nevatia, “Tracking of multiple, partially occluded humans based on static body part detection,” in 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06), vol. 1, pp. 951–958, June 2006.

[75] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, “Performance measures and a data set for multi-target, multicamera tracking,” in Computer Vision – ECCV 2016 Workshops (G. Hua and H. Jégou, eds.), (Cham), pp. 17–35, Springer International Publishing, 2016.

[76] J. M. S. Prewitt and M. L. Mendelsohn, “The analysis of cell images*,” Annals of the New York Academy of Sciences, vol. 128, no. 3, pp. 1035–1053, 1966.

[77] T. Y. Zhang and C. Y. Suen, “A Fast Parallel Algorithm for Thinning Digital Patterns,” Commun. ACM, vol. 27, pp. 236–239, Mar. 1984.
[78] P. Sahoo, S. Soltani, and A. Wong, “A survey of thresholding techniques,” *Computer Vision, Graphics, and Image Processing*, vol. 41, no. 2, pp. 233 – 260, 1988.

[79] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[80] Y. Li, C. Huang, and R. Nevatia, “Learning to associate: HybridBoosted multi-target tracker for crowded scene,” in 2009 *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2953–2960, June 2009.
A. Content

In these supplementary materials we provide: examples of the visual difference between the fish in the three splits, a more detailed explanation of the detectors, details on the benchmark pipeline parameters, a more detailed presentation of the relation between the proposed dataset complexity measures and tracking metrics, and the full set of tracking metrics for each step in the proposed benchmark pipeline.

B. Fish Examples

The visual appearance of the fish varies both within and between the splits as unique groups of fish have been used for each split. However, the two groups of fish used in the train and validation sets are from the same cohort, whereas the zebrafish in the test split are from a cohort of younger and smaller fish. We present a fish from each of the three splits in Figure 8 captured at the approximately same position in the water tank. The resolution of the zebrafish from the test split is significantly smaller compared to the fish in the training and validation splits, mainly due to the physical size of the fish.

C. Naive Object Detection

The Naive object detection algorithms presented briefly in the paper are described in more details in this section. The methods are inspired by Qian et al. [62] and their work on tracking zebrafish in 2D.

C.1. Pre-Processing

Initially the background image, without any fish, is estimated by taking the median of $N$ images sampled uniformly across the video. In the tests presented in the paper a set of 80 images were used to create the background image for each recording. All further processing is restricted within a defined region of interest based on the boundaries of the water tank and the level of water, in order to limit the processing time.

The video is downsampled by a factor of 2, in order to further decrease the processing time. Background subtraction is applied by calculating the absolute difference image, $|im - bg|$, choosing the max value across all color channels, and normalizing the image into the range $[0; 255]$. The resulting image is filtered with a $5 \times 5$ median filter to reduce noise.

C.2. Top-View Detection

The top-view is thresholded based on the assumption of the image histogram being bimodal, due to a near uniform bright background, and darker zebrafish. Therefore the intermodal approach of Prewitt et al. [76] can be utilized. The threshold is set to the middle point between the two modes in the image histogram. If the histogram is not bimodal, the frame is filtered with a $3 \times 3$ mean filter, until its histogram is bimodal.

The fish are detected by applying a skeletonization based approach. BLOBS are initially detected and filled, and the skeletonization approach of Zhang and Suen [77] is applied. The skeleton is analyzed in order to find the skeleton keypoints: head, tail, and junctions. This is done by convolving the frame with the kernel in Equation (8). All end points of the skeleton have a value of 116, 117, 118, or 131, while junctions have a value of 148, 149, 150, or 151. Values, like 132, that can represent both an end point or an arbitrary point on the skeleton are not considered.

Each keypoint is assigned a weight, $w$, consisting of the smallest eigenvalue of the covariance matrix of the BLOB coordinates extracted from a $20 \times 20$ window around the keypoint. The smallest eigenvalue is used, as it is assumed the variance will be smallest along the width of the head. This way the head keypoints will be assigned a larger $w$ than the tail keypoints, as the width of the zebrafish head is
usually larger than the width of the zebrafish tail. The \( w \) of the junction points are reduced by a factor of 2.5, as they are usually larger than endpoint keypoints.

Keypoints too close to each other are removed by applying non-maximal suppression (NMS). Each keypoint is assigned a bounding box of size \( w \times w \), and kept if the bounding box does not overlap with any other keypoint by more than NMS\(_{thresh}\)% of the area of the keypoint in focus. In case there is an overlap, only the keypoint with the largest \( w \) is kept. NMS\(_{thresh}\) was set to 50 in the conducted tests.

Finally, all keypoints with \( w < 1 \) are discarded. Per skeleton, the two keypoints furthest apart are determined, and the keypoint with largest \( w \) is kept. Furthermore, half of the additional extra keypoints with the largest \( w \) values are also kept in order to handle crossings and occlusions. Every keypoint ideally represent a zebrafish head and they are all kept as individual detections.

### C.3. Front-View Detection

The front frame cannot be thresholded based on the assumption of a bimodal distribution of the image histogram, as the stripes of the zebrafish results in a non-uniform appearance. Instead the frame is thresholded by finding the point maximizing the total entropy of the image [78]. The entropy is modeled as background and foreground entropy for each non-zero bin.

For bin \( k \) the background entropy, \( b_k \), is calculated by

\[
b_k = \sum_{i=0}^{k} \frac{h(i)}{h_c(i)} \log\left(\frac{h(i)}{h_c(i)}\right),
\]

where \( h \) is the normalized image histogram and \( h_c \) is the cumulative histogram of \( h \). The foreground entropy, \( w_k \), is calculated by

\[
w_k = \sum_{i=k+1}^{255} \frac{h(i)}{1-h_c(i)} \log\left(\frac{h(i)}{1-h_c(i)}\right),
\]

resulting in the two entropy sets \( B = \{b_0, b_1, ..., b_{255}\} \) and \( W = \{w_0, w_1, ..., w_{255}\} \) which are combined into

\[
E = B + W,
\]

and the threshold, \( t \), is then finally determined by

\[
t = \arg\max_x E(x).
\]

The BLOBs in the thresholded image are found through simple Connected Component Analysis (CCA). Only the \( 2N_{fish} \) largest BLOBs are kept, as long as the area of the BLOBs are larger than a predefined threshold. This allows for potential reflections being detected alongside the true detection of the fish. Otherwise, if only the \( N_{fish} \) largest BLOBs are considered for further analysis, it is possible that reflections are considered at the expense of real fish.

As it is not known where the head of the fish is, two points are saved as proxies for the head, together with the center-of-mass. If the width of the BLOB bounding box is larger than the height, the proxy points are \((\min(x), \mu(y))\) and \((\max(x), \mu(y))\), where \( x \) and \( y \) are the coordinates of the BLOB pixels, and \( \mu \) is the arithmetic mean function. Otherwise, the proxy points are \((\mu(x), \min(y))\) and \((\mu(x), \max(y))\). The center-of-mass is further used for constructing 2D tracklets, as it is the most stable of the three proxy points.

### D. FRCNN-H Object Detection

Two Faster R-CNN networks were trained to detect the zebrafish heads in each view, respectively. The official PyTorch implementation of Faster R-CNN with a ResNet50
Table 4: Statistics of the reprojection error and speed of the zebrafish, for each split, based on the ground truth annotations.

|       | Property      | Mean   | Std. Dev. | Median |
|-------|---------------|--------|-----------|--------|
| Trn   | Reproj. Error [px] | 8.03   | 5.26      | 7.59   |
|       | Speed [cm/s²]   | 2.13   | 2.32      | 1.54   |
| Val   | Reproj. Error [px] | 6.22   | 4.50      | 5.43   |
|       | Speed [cm/s²]   | 2.02   | 2.37      | 1.41   |
| Tst   | Reproj. Error [px] | 4.36   | 3.27      | 3.69   |
|       | Speed [cm/s²]   | 2.11   | 1.93      | 1.58   |

Based Feature Pyramid Network [79] backbone was utilized in both cases. The model was pretrained on the COCO train2017 dataset. The network was fine-tuned for 30 epochs, using stochastic gradient descent with momentum. A learning rate of 0.005, momentum of 0.9, weight decay of 0.0005, and batch size of 8 was used. The learning rate was “warmed up” during the first epoch, linearly interpolating the learning rate from 0.001 to 0.005. Each model was trained on an RTX 2080 TI.

E. Pipeline Parameters

The full system pipeline has a set of parameters, which needs to be manually set. The parameters were chosen based on empirical investigation on the training data and they are shown in Table 3. The front-view distance threshold, $\delta_F$, has two different values, depending on the method applied. For the Naive method it is set to 0.5, as the distance is measured in standard deviations, whereas for the FRCNN-H method it is set to 15, where the distance is measured in pixels.

Furthermore, during the association of 2D tracklets between views, a set of exponential distributions are utilized. These distributions are parameterized using the mean of the reprojection error of the ground truth training annotations and the average speed of the zebrafish in the training split. This builds on the assumption that these parameters generalize across the different splits. As shown in Table 4, the average speed of the zebrafish for each split are within 0.1 cm/s, which can be seen as a negligible difference, whereas the mean reprojection error is 2 – 4 pixels larger in the training and validation splits than in the testing split. The cause of this difference is hypothesized to be the larger amount and duration of occlusions in the training and validation splits, which can introduce some uncertainty during the annotation phase. The training split has the largest reprojection error, which means the utilized exponential distribution penalize larger reprojection errors less harshly. However, as the mean training reprojection error is still low, it is still deemed fitting for the task.

F. Tracking Metrics and Results

When evaluating tracking tasks, a wide suite of metrics are often used, each expressing different properties of the task. Commonly, the MOTA metric is used as a representative metric for the overall performance of the tracker, as it incorporates false positives and negatives, as well as the amount of identity swaps. However, as shown by Carr and Collins [72] the MOTA metric is not guaranteed to be robust at all times. We evaluate the generated tracks using the collection of metrics from the MOT challenge, consisting of the CLEAR MOT [73], mostly tracked/lost metric [74], and identity based metrics of Ristani et al. [75], as well as using the MTBF metric proposed by Carr and Collins [72].

A description of each of the used metrics are presented in Table 5. Each step of the pipeline has been evaluated according to the previous mentioned metrics. Specifically the following steps have been evaluated:

- 2D tracklets after 2D Tracklet Construction (for top-view see Table 6 and for front-view see Table 7)
- 2D tracklets after 2D Tracklet Association Between Views (for top-view see Table 8 and for front-view see Table 9)
- 3D tracklets after 2D Tracklet Association Between Views (see Table 10)
- 3D tracks after 3D Tracklet Association (see Table 11)

For the 2D tracklets, a distance threshold of 20 px is used.
The two views are not equal when it comes to occlusions. Due to the social behavior of zebrafish they tend to swim in the same horizontal layer of the water column. This is indicated by the graph in Figure 9 where the occlusion lengths are displayed with respect to the proposed complexity measure, $\Psi$. It should be noted that $\Psi$ is calculated per view and not as a mean between the two, as in the main article. Data points are presented for all the recordings from both views, except Tst1 as it only contains a single fish and therefore has a complexity measure of zero. The dashed lines illustrate the mean occlusion lengths and shows that there is a significant difference between the two views. In general, the occlusion events seems longer in the front-view recordings.

The MOTA of the hypothetical oracle tracker is plotted against the proposed complexity measure in Figure 10. MOTA is calculated for the top- and front-view 2D tracklets as presented in Table 6 and Table 7, respectively. The tracking performance of the oracle tracker correlates well with the complexity in both views. Notice the significant difference in performance of the tracker in the two views; the mean scores are $\text{MOTA}_{\text{top}} = 86.0\%$ and $\text{MOTA}_{\text{front}} = 75.5\%$ as illustrated by the two dashed lines.
MOTA Higher \([-\infty, 100]\) Overall tracking accuracy based on ID-swaps, and false negatives/positives \([73]\)

MOTP Lower \([0, \infty]\) Distance between predicted position and ground truth \([73]\)

Prc. Higher \([0, 100]\) Precision: The percentage of correctly predicted positions

Rcll. Higher \([0, 100]\) Recall: The percentage of ground truth positions detected

ID-Prc. Higher \([0, 100]\) ID-Precision: The percentage of correctly identified predicted positions \([75]\)

ID-Rcll. Higher \([0, 100]\) ID-Recall: The percentage of correctly identified ground truth positions \([75]\)

ID-F1 Higher \([0, 100]\) F1-score of the predicted identities \([75]\)

FP Lower \([0, \infty]\) Amount of incorrectly predicted positions

FN Lower \([0, \infty]\) Amount of ground truth positions missed

MT Higher \([0, 100]\) Amount of ground truth tracks which are 80% or more correctly tracked \([74]\)

ML Lower \([0, 100]\) Amount of ground truth tracks which are 20% or less correctly tracked \([74]\)

ID Sw. Lower \([0, \infty]\) Total amount of identity switches \([80]\)

Frag. Lower \([0, \infty]\) Total amount of track fragmentations

MTBFs Higher \([0, \infty]\) Mean time between ID-switches or false negatives/positives \([72]\)

MTBFm Higher \([0, \infty]\) Monotonic version of MTBFs \([72]\)

Table 5: Description of the full collection of utilized tracking metrics.
Table 8: Full metric evaluation of the 2D tracking in the top-view, after 3D tracklet association, on all sequences.
| Method | MOTA | MOTP | Prc. | Rcll. | ID-Rcll. | ID-Prc. | ID-F1 | FP | FN | MT | ML | ID Sw. | Frag. | MTBF_s | MTBF_m |
|--------|------|------|------|------|---------|---------|-------|----|----|----|----|-------|-------|--------|--------|
| Naive  | 41.1%| 0.198| 85.0 | 50.5 | 9.1     | 15.3    | 11.4  | 1283| 7118| 0   | 60  | 586   | 12.344| 6.172  |
| FRCNN-H| 73.7%| 0.066| 96.9 | 76.3 | 18.8    | 23.8   | 21.0  | 352 | 3407| 0   | 29  | 215   | 50.548| 25.333 |
| Oracle | 44.8%| 0.000| 100.0| 46.2 | 2.4     | 5.3    | 3.3   | 0   | 7738| 0   | 0   | 0     | 32.539| 16.190 |
| Naive  | 40.3%| 0.169| 83.5 | 50.5 | 25.5    | 42.1   | 31.4  | 442 | 2218| 0   | 12  | 134   | 35.725| 18.039 |
| FRCNN-H| 63.5%| 0.064| 92.1 | 69.7 | 37.1    | 49.1   | 42.3  | 269 | 1358| 0   | 9   | 67    | 43.541| 21.867 |
| Oracle | 36.7%| 0.000| 100.0| 83.2 | 24.3    | 29.2   | 26.6  | 0   | 1658| 0   | 2   | 10    | 74.900| 37.450 |
| Naive  | 39.7%| 0.061| 95.6 | 79.4 | 79.4    | 95.6   | 86.7  | 66  | 371 | 0   | 38  | 93    | 43.333| 20.409 |
| FRCNN-H| 50.1%| 0.546| 99.9 | 50.4 | 53.7    | 36.0   | 13    | 2209| 0   | 23  | 91    | 54.691| 27.345 |
| Oracle | 65.7%| 0.000| 100.0| 66.8 | 26.8    | 40.1   | 32.2  | 1429| 0   | 50  | 50    | 900.000| 900.000|
| Naive  | 59.7%| 0.063| 92.2 | 65.4 | 48.5    | 68.3   | 56.7  | 200 | 1244| 0   | 5   | 86    | 26.682| 13.341 |
| FRCNN-H| 51.4%| 0.062| 96.3 | 74.4 | 61.4    | 79.5   | 69.3  | 103 | 919 | 0   | 4   | 36    | 70.342| 36.122 |
| Oracle | 74.1%| 0.000| 100.0| 75.1 | 12.8    | 17.0   | 14.6  | 894 | 0   | 36  | 36    | 71.000| 36.459 |
| Naive  | 28.4%| 0.204| 79.8 | 39.1 | 23.2    | 46.8   | 31.1  | 47  | 2764| 0   | 9   | 91    | 18.122| 9.015  |
| FRCNN-H| 61.3%| 0.070| 95.4 | 64.6 | 33.7    | 49.7   | 40.2  | 143 | 1607| 1   | 9   | 60    | 45.123| 22.562 |
| Oracle | 42.4%| 0.000| 100.0| 43.8 | 13.3    | 30.3   | 18.5  | 0   | 2552| 0   | 64  | 64    | 28.812| 14.618 |
| Naive  | 77.6%| 0.152| 96.0 | 80.9 | 80.9    | 96.0   | 87.8  | 30  | 171 | 0   | 28  | 25.000| 12.500| 6.812  |
| FRCNN-H| 53.0%| 0.105| 100.0| 30.2 | 30.2    | 100.0  | 46.4  | 0   | 625 | 0   | 15  | 16.838| 9.218  |
| Oracle | 100.0%| 0.000| 100.0| 100.0| 100.0   | 100.0  | 100.0| 0  | 0   | 0   | 0   | 0     | 900.000| 900.000|

Table 9: Full metric evaluation of the 2D tracking in the front-view, after 3D tracklet association, on all sequences.
| Method | MOTA | MOTP | Prc. | Rcll. | ID-Rcll. | ID-Prc. | ID-F1 | FP | FN | MT | ML | ID Sw. | MTBF | MTBF_
|--------|------|------|-----|------|---------|---------|-------|----|----|----|----|-------|-------|-------|
| Naive | 41.4 | 0.198 | 85.8 | 50.4 | 28.5 | 48.3 | 35.9 | 1250 | 7133 | 0 | 0 | 40 | 573 | 12.597 | 6.298 |
| FRCNN-H | 73.8 | 0.066 | 96.9 | 76.3 | 44.0 | 55.9 | 49.2 | 352 | 3407 | 0 | 0 | 14 | 216 | 50.317 | 25.216 |
| Oracle | 46.2 | 0.000 | 100.0 | 46.2 | 46.2 | 100.0 | 63.2 | 0 | 7738 | 0 | 0 | 0 | 202 | 32.539 | 16.190 |
| Naive | 40.4 | 0.170 | 83.8 | 50.3 | 32.5 | 54.2 | 40.7 | 436 | 2228 | 0 | 0 | 7 | 135 | 16.121 | 7.947 |
| FRCNN-H | 63.5 | 0.064 | 92.1 | 69.7 | 40.9 | 54.1 | 46.6 | 269 | 1359 | 1 | 0 | 7 | 66 | 44.028 | 22.170 |
| Oracle | 37.9 | 0.000 | 100.0 | 37.9 | 37.9 | 100.0 | 55.0 | 0 | 2784 | 0 | 0 | 0 | 53 | 29.328 | 14.175 |
| Naive | 59.6 | 0.163 | 92.2 | 65.3 | 50.8 | 71.7 | 59.4 | 199 | 1248 | 0 | 0 | 3 | 82 | 27.905 | 13.870 |
| FRCNN-H | 71.5 | 0.062 | 96.3 | 74.4 | 71.7 | 92.8 | 80.9 | 103 | 920 | 0 | 0 | 2 | 35 | 72.216 | 37.111 |
| Oracle | 75.1 | 0.000 | 100.0 | 75.1 | 75.1 | 100.0 | 85.8 | 0 | 894 | 0 | 0 | 0 | 36 | 71.000 | 36.459 |
| Naive | 28.5 | 0.204 | 78.9 | 39.1 | 23.2 | 46.8 | 31.1 | 476 | 2764 | 0 | 0 | 7 | 93 | 18.122 | 9.015 |
| FRCNN-H | 61.3 | 0.070 | 95.4 | 64.6 | 40.2 | 59.4 | 48.0 | 143 | 1607 | 0 | 0 | 5 | 60 | 45.123 | 22.562 |
| Oracle | 43.8 | 0.000 | 100.0 | 43.8 | 43.8 | 100.0 | 60.9 | 0 | 2552 | 0 | 0 | 0 | 64 | 28.812 | 14.618 |
| Naive | 77.6 | 0.152 | 96.0 | 80.9 | 80.9 | 96.0 | 87.8 | 30 | 171 | 1 | 0 | 0 | 28 | 25.000 | 12.500 |
| FRCNN-H | 30.2 | 0.105 | 100.0 | 30.2 | 30.2 | 100.0 | 46.4 | 0 | 625 | 0 | 0 | 0 | 15 | 16.938 | 8.212 |
| Oracle | 100.0 | 0.000 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 900.000 |
| Naive | 71.0 | 0.138 | 96.9 | 80.2 | 80.2 | 96.9 | 87.7 | 46 | 353 | 1 | 0 | 0 | 44 | 31.022 | 15.856 |
| FRCNN-H* | 12.5 | 0.000 | 100.0 | 12.5 | 12.5 | 100.0 | 60.2 | 0 | 1677 | 0 | 2 | 2 | 17 | 5.421 | 2.641 |
| Oracle | 81.6 | 0.000 | 100.0 | 81.6 | 81.6 | 100.0 | 89.9 | 0 | 328 | 2 | 0 | 0 | 25 | 53.778 | 27.396 |
| Naive | 39.7 | 0.168 | 80.0 | 53.1 | 43.4 | 63.4 | 52.2 | 588 | 2079 | 0 | 0 | 7 | 185 | 12.400 | 6.249 |
| FRCNN-H | 40.2 | 0.099 | 98.0 | 41.2 | 29.8 | 70.9 | 41.9 | 37 | 2609 | 0 | 0 | 7 | 115 | 15.217 | 7.577 |
| Oracle | 67.8 | 0.000 | 100.0 | 67.8 | 67.8 | 100.0 | 80.8 | 0 | 1427 | 1 | 0 | 0 | 50 | 54.691 | 28.112 |
| Naive | 48.3 | 0.153 | 86.9 | 57.1 | 48.5 | 73.8 | 58.5 | 768 | 3829 | 0 | 0 | 11 | 268 | 18.313 | 9.075 |
| FRCNN-H* | 12.8 | 0.000 | 97.0 | 26.3 | 14.1 | 52.0 | 22.2 | 72 | 6571 | 0 | 0 | 3 | 32 | 225 | 9.996 |
| Oracle | 66.6 | 0.000 | 100.0 | 66.6 | 66.6 | 100.0 | 79.9 | 0 | 2982 | 1 | 0 | 0 | 119 | 46.031 | 23.105 |

Table 11: Full metric evaluation of the final 3D tracks generated from the 3D track association, on all sequences. An * indicates that the method did not complete the 3D Tracklet Association step for the sequence, as there were not enough concurrent 3D tracklets at any one point in the sequence. In this case, the 3D tracklet results are reported.