

Tuna Nutriment Tracking using Trajectory Mapping in Application to Aquaculture Fish Tank

Hilmil Pradana

Graduate School of Life Science and Systems Engineering
Kyushu Institute of Technology

Fukuoka, Japan

Email: pradana.muhamad-hilmil505@mail.kyutech.jp

Keiichi Horio

Graduate School of Life Science and Systems Engineering
Kyushu Institute of Technology

Fukuoka, Japan

Email: horio@brain.kyutech.ac.jp

Abstract—The cost of fish feeding is usually around 40 percent of total production cost. Estimating a state of fishes in a tank and adjusting an amount of nutriment play an important role to manage cost of fish feeding system. Our approach is based on tracking nutriment on videos collected from an active aquaculture fish farm. Tracking approach is applied to acknowledge movement of nutriment to understand more about the fish behavior. Recently, there has been increasing number of researchers focused on developing tracking algorithms to generate more accurate and faster determination of object. Unfortunately, recent studies have shown that efficient and robust tracking of multiple objects with complex relations remain unsolved. Hence, focusing to develop tracking algorithm in aquaculture is more challenging because tracked object has a lot of aquatic variant creatures. By following aforementioned problem, we develop tuna nutriment tracking based on the classical minimum cost problem which consistently performs well in real environment datasets. In evaluation, the proposed method achieved 21.32 pixels and 3.08 pixels for average error distance and standard deviation, respectively. Quantitative evaluation based on the data generated by human annotators shows that the proposed method is valuable for aquaculture fish farm and can be widely applied to real environment datasets.

Index Terms—Productivity, Fish Feeding, Nutriment, Tracking Algorithm, Real Environment Datasets

I. INTRODUCTION

Aquaculture is the one of farming type in which aquatic creatures require acceptable environment for living habitat and availability nutriment to increase productivity and sustain healthy growth [1]–[4]. Within current requiring acceptable habitat, water quality is also a vital component to enlarge fish fertility rate [3]–[6]. Water quality can be obtained by cleaned often and give optimal amount of nutriment. Increasing number of nutriment can affect a lot of foods wasted in the water and quality of water occurs highly polluted. On the other hand, reducing feeding will lead starvation and drop fish quality. So that, management of nutriment delivered is vital component to balance productivity rate [7], [8].

The cost of fish feeding is usually around 40 percent of total production cost [9]–[11]. Estimating a state of fishes in a tank and adjusting an amount of nutriment play an important role to manage cost of fish feeding system. It is applied to control the amount of nutriment and realizes the fish behavior in tank. Lately, application to monitor fish behavior

has been adopted by a telemetry-based approach [12], [13] and a computer vision(CV)-based approach [14]–[20].

A telemetry-based approach is a technique attaching an external transmitter by mounting, or surgical implantation in the peritoneal cavity [12]. Attaching a transmitter in each fish will spend higher cost and its transmitter can only set in large fish. When their fishes had been farmed, attachment will always be given to new fishes. On the other hand, CV-based approach studies are not required complexity analysis such as ripple activity and tracking analysis in which, small number of fishes and small tanks with special environment assist creating result. Tracking approach is applied to acknowledge movement of nutriment to understand more about the fish behavior. Fish behaviors can be obtained by combination between tracking analysis and ripple activity. Then, these fish behaviors can be a decision to start and stop fish feeding machine by understanding of ripple activity after giving several nutriment. By explaining of fish behavior, tracking nutriment is important and it is required to analyze the complexity data in real environment.

Recently, there has been increasing number of researchers focused on developing tracking algorithms to generate more accurate and faster determination of object. Tracking can be represented as a graph problem which can solved by a frame-by-frame [21]–[24] or track-by-track [25], [26]. Interpretation of tracking problems with data association mostly uses a graph, where each detection is called as vertex, and each edge is pointing any possible link among them out as object tracked. Data association can be declared as minimum cost problem [27]–[30] with learning cost problem [31] or motion pattern maps [32]. Alternative formulations to solve optimization problems is minimum clique problem [33] and lifted multicut problem [34] where its formulations follow body pose layout to obtain estimated model. Recently, efficient and robust tracking of multiple objects with complex relations remain unsolved. Hence, focusing to develop tracking algorithm in aquaculture is more challenging because tracked object has a lot of aquatic variant creatures. By summarizing aforementioned problems, we proposed tuna nutriment tracking based on the classical minimum cost problem [28]–[30] where each detection calculates minimum distance among them and creates a trajectory to be tracked line. By collaborating with

an active aquaculture fish farm, we develop tuna nutriment tracking using trajectory mapping. A video camera is placed above the boat with a highly disturbance of ocean wave and many dense nutriments. The camera captures between ocean surface and fish feeding machine. After that, videos transfer to a computer for further analysis the behavior of fish.

The aim of this research is tracking approach to acknowledge the behavior of tuna. For next, it can be useful to improve the production profit in fish farms by controlling the amount of nutriment in optimal rate.

To summarize, we make the following **contributions**:

- We propose tuna nutriment tracking based on trajectory mapping which can perform well as well as human annotator results.
- We propose a new novel small nutriment tracking method with collecting information of leading line into ripple.
- We show significantly improvement result of trajectory mapping in real environment datasets.

II. THE PROPOSED METHOD - TRAJECTORY MAPPING

Our formulation is based on the classical minimum cost problem where each detection calculates minimum distance among them and creates a trajectory to be tracked line. In order to provide some background and formally introduce our approach, we start by providing flowchart and algorithm of tuna nutriment tracking. We then explain how the proposed method works to real environment. The proposed trajectory mapping contains a data normalization process, tuna nutriment detection and tuna nutriment tracking. The system flowchart of the proposed method is shown in Fig. 1, and the algorithm of the proposed trajectory mapping is represented in Fig. 2 where D and \mathbf{T}_f are input video and trajectory of time-ordered tuna nutriment, respectively.

A. Data Normalization

For data normalization, image stabilization is applied to reduce a hand-held camera and ocean waves. Image stabilization is created by transformation from previous to current frame using optical flow for all frames. [35] accumulates rigid transformation χ to obtain linked between frame L . New rigid transformation χ_ϕ in frame ϕ can be written as:

$$\chi_\phi = \chi_{\phi-1} + \left(\frac{1}{\gamma} \sum_{\tau=\phi-\gamma}^{\phi+\gamma} L_\tau\right) - L_{\phi-1}, \quad (1)$$

$$\hat{D} = \chi_\phi\{D\},$$

where \hat{D} is output video after applied image stabilization and γ is smoothing radius where the radius is number of frames used for smoothing and defined by 30.

B. Tuna Nutriment Detection

The idea of tuna nutriment detection is to produce boundary box in each nutriment associated in tracking method. In implementation of tuna nutriment detection, YOLOv3 [36] accumulates bounding box of tuna nutriment prediction $B =$

$(\hat{x}, \hat{y}, \hat{w}, \hat{h})$ by training model with bounding box $P = (p_x, p_y, p_w, p_h)$ of ground truth data where $p_x, p_y, p_w,$ and p_h are centroid x , centroid y , width, and height of bounding box in ground truth data, respectively. ς_x and ς_y represent the absolute location of the top-left corner of the current grid cell. w and h are the absolute width and height to the whole image. Bounding box of tuna nutriment prediction B can be defined as:

$$\begin{aligned} \hat{x} &= \delta(p_x) + \varsigma_x \\ \hat{y} &= \delta(p_y) + \varsigma_y \\ \hat{w} &= e^{p_w} * w \\ \hat{h} &= e^{p_h} * h \end{aligned} \quad (2)$$

where δ is model followed by [36].

C. Tuna Nutriment Tracking

In order to represent tracking of tuna nutriment, introducing how to collect set of tuna nutriment prediction corresponding to time-ordered path in the graph is important. We are given $\sigma^c(n) = \{C_1, C_2, C_3, \dots, C_n\}$ as input centroid of tuna nutriment predictions where n is the total number of nutriment for all frames of video \hat{D} . Each tuna nutriment prediction is represented by $C_n = \{\hat{x}_n^c, \hat{y}_n^c\}$. Definition of a trajectory is denoted as centroid of time-ordered tuna nutriment predictions $\mathbf{T}_f(n_f) = \{C_{f_1}, C_{f_2}, C_{f_3}, \dots, C_{f_{n_f}}\}$ where n_f is the number of detections formed by trajectory f . So that, $\varrho = \{n_1, n_2, n_3, \dots, n_{n_f}\}$ can be denoted as the total of number of nutriments appearing in every time-ordered trajectory $\mathbf{T}_f(n_f)$.

1) *Problem Statement*: The problem can be represented with an undirected graph $G = (V, E)$, where $V := \{1, \dots, n\}, E \subset V^2$, and each node $f \in V$ denotes a unique detection $C_f \in \sigma^c$. The task of dividing the set of tuna nutriment predictions into trajectories can be observed as grouping nodes in graph. Fig. 3 shows that each trajectory $\mathbf{T}_f(n_f) = \{C_{f_1}, C_{f_2}, C_{f_3}, \dots, C_{f_{n_f}}\}$ in the scene can be mapped into a group of nodes $\{(f_1, f_2), (f_2, f_3), \dots, (f_{n-1}, f_{n_f})\}$. To produce each $C_{f_{[1, n_f]}}$, trajectory mapping is applied in next section.

In two-dimensional trajectory, the component of trajectory is divided by horizontal and vertical direction. In vertical direction, acceleration is constant and has quadratic function. Trajectory mapping applies the idea of acceleration and chooses quadratic function as basis.

To produce quadratic function $y^c = a_3^c x^2 + a_2^c x + a_1^c$ as a result of trajectory \mathbf{T}_f , we apply polynomial fitting [37] defined by calculation of $\hat{x}_{n_f}^c$ to form Vandermonde matrix V with 3 columns as results of \mathbf{a}^c .

$$\begin{bmatrix} 1 & \hat{x}_1^c & (\hat{x}_1^c)^2 \\ 1 & \hat{x}_2^c & (\hat{x}_2^c)^2 \\ \vdots & \vdots & \vdots \\ 1 & \hat{x}_{n_f}^c & (\hat{x}_{n_f}^c)^2 \end{bmatrix} \begin{bmatrix} a_1^c \\ a_2^c \\ a_3^c \end{bmatrix} = \begin{bmatrix} \hat{y}_1^c \\ \hat{y}_2^c \\ \vdots \\ \hat{y}_{n_f}^c \end{bmatrix} \quad (3)$$

(3) can be inverted directly. To yield the solution vector \mathbf{a}^c , it can be defined as:

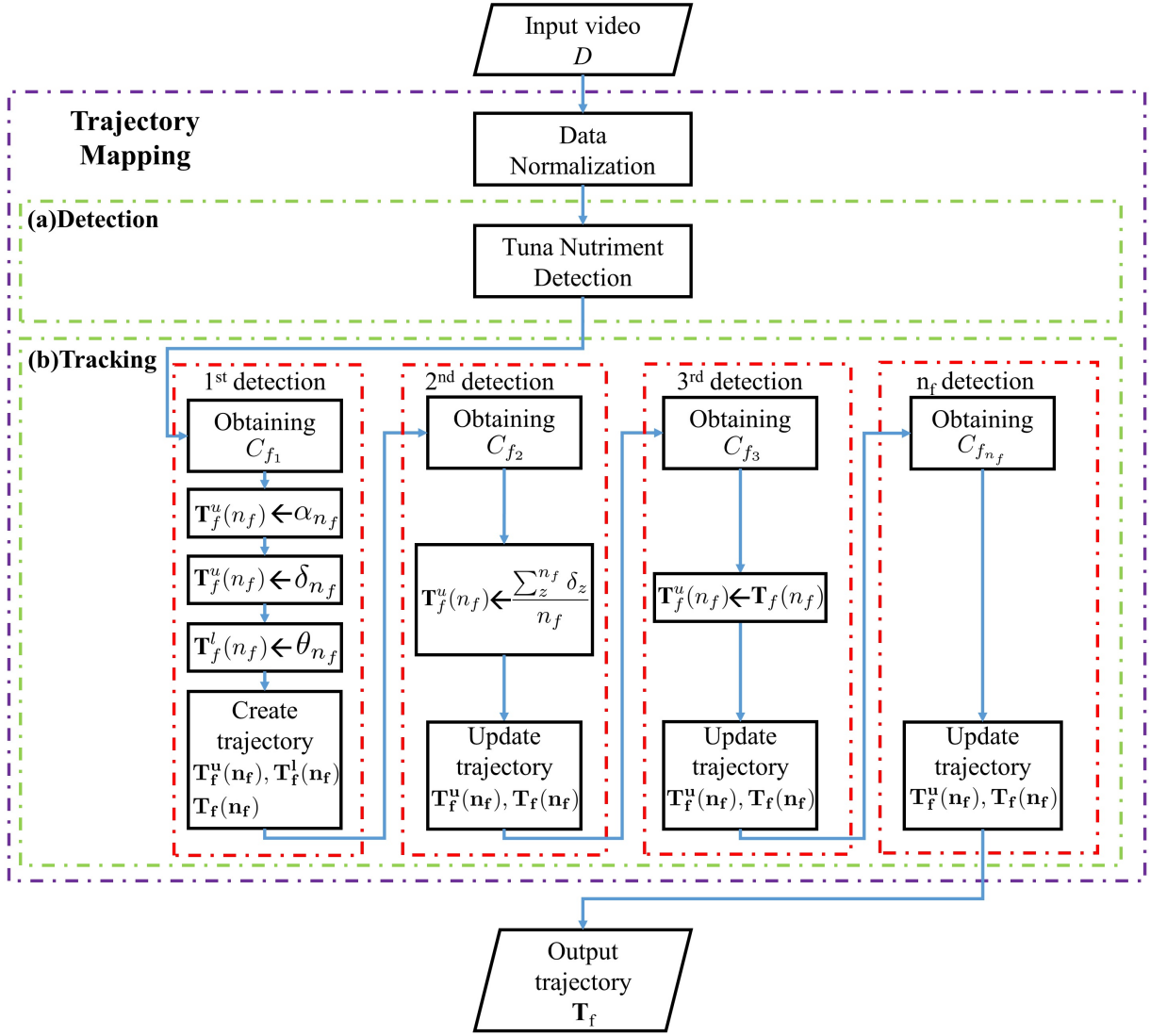


Fig. 1: Flowchart of the proposed trajectory mapping. The input video is received and applied image stabilization as data normalization. (a) Creating model for tuna nutriment detection using YOLOv3 [36] and obtaining bounding box for each tuna nutriment prediction in all frames of a video. (b) Tuna nutriment prediction as input tracking approach to be initialization for $n_f = 1$ detection. After that, we obtain upper and lower limit trajectory T_f^u , T_f^l and the maximum height of all tuna nutriment predictions δ_1 . Next, we find the value of C_{f_2} using the shortest distance between C_{f_1} and all tuna nutriment predictions in n_f appearing in inside area between $T_f^u(1)$ and $T_f^l(1)$. Its process is repeatedly until n_f . Next, we obtain final trajectory as tracking result.

$$\mathbf{a}^c = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}, \quad (4)$$

2) Tuna Nutriment Predictions $C_{f_{n_f}}$ where $n_f = 1$ as Initialization Point Detection: Tuna nutriment predictions C_{f_1} are obtained from every tuna nutriment prediction C_n in around cutting area of w . To define cutting area, we use centroid \hat{x}_{f_1} as component of C_{f_1} by thresholding in w which is defined as:

$$w * \gamma \leq \hat{x}_{f_1} \leq w, \quad (5)$$

where γ is an input parameter and empirically defined as 0.9.

Direction of nutriment is calculated by leading nutriment to ripple area around sea levels. We are given a pair set of ripple area detection $\sigma^r(n_f) = \{(R_{11}, R_{12}), (R_{21}, R_{22}), (R_{31}, R_{32}), \dots, (R_{n_f1}, R_{n_f2})\}$ as time-ordered ripple predictions in number of detections n_f . Each ripple prediction is represented by $R_{n_f1,2} = \{\hat{x}_{n_f}^r, \hat{y}_{n_f}^r, \hat{w}_{n_f}^r, \hat{h}_{n_f}^r\}$. We divide component of ripple prediction to be an area of top-left $\alpha_{n_f} = (\hat{x}_{n_f}^\alpha, \hat{y}_{n_f}^\alpha)$ and bottom right $\theta_{n_f} = (\hat{x}_{n_f}^\theta, \hat{y}_{n_f}^\theta)$ of ripple detection by following:

Algorithm 1: Trajectory mapping algorithm

Input : D : a fish feeding video data.
Auxiliary methods: Data normalization, tuna nutrient detection, tuna nutrient tracking.
Output : Trajectory T_f

- 1 input a fish feeding video data D .
- 2 use image stabilization as data normalization
 $\hat{D} = \chi_\phi\{D\}$.
- 3 **for** $\varpi \leftarrow 1$ to n_f **do**
- 4 calculate centroid of tuna nutrient detection $\sigma^c(\varpi)$ and ripple area detection $\sigma^r(\varpi)$.
- 5 create an area of top-left $\alpha_\varpi = (\hat{x}_\varpi^\alpha, \hat{y}_\varpi^\alpha)$ and bottom right $\theta_\varpi = (\hat{x}_\varpi^\theta, \hat{y}_\varpi^\theta)$ of ripple detection.
- 6 **if** $\varpi = 1$ **then**
- 7 create initialization of upper limit trajectory $T_f^u(\varpi) = \{C_{f_1}, \delta_\varpi, \alpha_\varpi\}$ and lower limit trajectory $T_f^l(\varpi) = \{C_{f_1}, \theta_\varpi\}$.
- 8 **end**
- 9 **else if** $\varpi = 2$ **then**
- 10 update upper limit trajectory
 $T_f^u(\varpi) = \{C_{f_1}, C_{f_2}, \frac{\sum_{z=1}^{\varpi} \delta_z}{n_f}, \alpha_\varpi\}$.
- 11 **end**
- 12 **else if** $\varpi = 3$ **then**
- 13 update upper limit trajectory
 $T_f^u(\varpi) = \{C_{f_1}, C_{f_2}, C_{f_3}, \frac{\sum_{z=1}^{\varpi} \delta_z}{\varpi}, \alpha_\varpi\}$.
- 14 set trajectory of tuna nutrient prediction $T_f(\varpi) = T_f^u(\varpi)$.
- 15 **end**
- 16 **else**
- 17 update trajectory of tuna nutrient prediction
 $T_f(\varpi) = \{C_{f_1}, C_{f_2}, \dots, C_{f_\varpi}, \frac{\sum_{z=1}^{\varpi} \delta_z}{\varpi}, \alpha_\varpi\}$.
- 18 **end**
- 19 **end**

Fig. 2: Algorithm of the proposed trajectory mapping. Iterative method are applied to improve trajectory T_f by collecting each centroid of nutrient in every frame.

$$\begin{aligned}\hat{x}_{n_f}^\alpha &= \hat{x}_{n_f}^r - \frac{\hat{w}_{n_f}^r}{2}, \\ \hat{y}_{n_f}^\alpha &= \hat{y}_{n_f}^r - \frac{\hat{h}_{n_f}^r}{2}, \\ \hat{x}_{n_f}^\theta &= \hat{x}_{n_f}^r + \frac{\hat{w}_{n_f}^r}{2}, \\ \hat{y}_{n_f}^\theta &= \hat{y}_{n_f}^r + \frac{\hat{h}_{n_f}^r}{2},\end{aligned}$$

To obtain more feature, we need to know possibly coverage area for possibly nutrient appearing in next frame by creating upper and lower limit trajectory T_f^u and T_f^l , respectively.

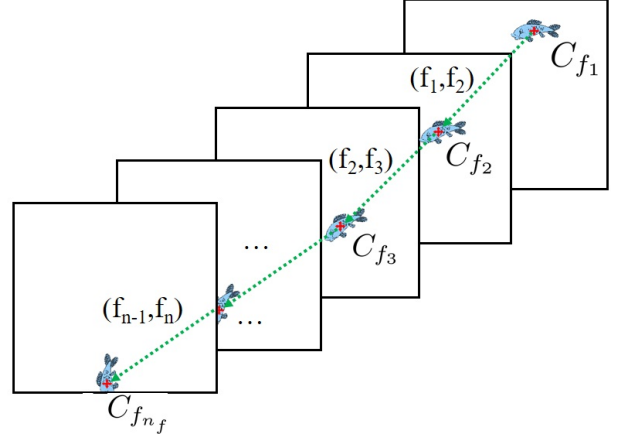


Fig. 3: Visualization of trajectory of tuna nutrient predictions $T_f = \{C_{f_1}, C_{f_2}, C_{f_3}, \dots, C_{f_{n_f}}\}$ in which every tuna nutrient predictions is connected by node $\{(f_1, f_2), (f_2, f_3), \dots, (f_{n-1}, f_{n_f})\}$.

Upper and lower limit trajectory $T_f^u(n_f)$ and $T_f^l(n_f)$ formed by trajectory f are initialized by following:

$$\begin{aligned}T_f^u(n_f) &= \{C_{f_1}, \delta_{n_f}, \alpha_{n_f}\}, \\ T_f^l(n_f) &= \{C_{f_1}, \theta_{n_f}\},\end{aligned}\quad (7)$$

where δ is the maximum height of all nutrient detections in n_f . (7) can be simplify by substituting $n_f = 1$ to be:

$$\begin{aligned}T_f^u(1) &= \{C_{f_1}, \delta_1, \alpha_1\}, \\ T_f^l(1) &= \{C_{f_1}, \theta_1\},\end{aligned}\quad (8)$$

where $\delta_1 = (\frac{\hat{x}_1^\alpha + w}{2}, \min_{1 \leq z \leq n_1} \hat{y}_z^c)$.

3) *Tuna Nutrient Predictions $C_{f_{n_f}}$ where $n_f = 2$:* To be a candidate of C_{f_2} , we use all tuna nutrient predictions n appearing in the inside of area between $y^u = a_3^u x^2 + a_2^u x + a_1^u$ and $y^l = a_3^l x^2 + a_2^l x + a_1^l$. Vector \mathbf{a}^u and \mathbf{a}^l are produced by calculating $T_f^u(n_f - 1)$ and $T_f^l(n_f - 1)$ with Vandermonde matrix shown in (3) and (4), respectively. Given $\sigma^\kappa(\mu) = \{\kappa_{1_{n_f}}, \kappa_{2_{n_f}}, \kappa_{3_{n_f}}, \dots, \kappa_{\mu_{n_f}}\}$ is a set of candidate $C_{f_{n_f}}$. $C_{f_{n_f}}$ is defined by the nutrient predictions which have shortest distance denoted by:

$$\begin{aligned}C_{f_{n_f}} &= \arg \min_{\mu} (\mathbf{Z}(f_{n_f-1}) - \sigma^\kappa(\mu))^T \\ &\quad (\mathbf{Z}(f_{n_f-1}) - \sigma^\kappa(\mu))\end{aligned}\quad (9)$$

where $\mathbf{Z}(f_{n_f-1}) = \{C_{f_{n_f-1}}^1, C_{f_{n_f-1}}^2, C_{f_{n_f-1}}^3, \dots, C_{f_{n_f-1}}^\mu\}$. (9) can be simplify to be;

$$C_{f_2} = \arg \min_{\mu} (\mathbf{Z}(1) - \sigma^\kappa(\mu))^T (\mathbf{Z}(1) - \sigma^\kappa(\mu)),\quad (10)$$

Updating upper trajectory $T_f^u(n_f)$ can be defined as:

$$T_f^u(n_f) = \{C_{f_1}, C_{f_2}, \frac{\sum_{z=1}^{n_f} \delta_z}{n_f}, \alpha_{n_f}\},\quad (11)$$

4) *Tuna Nutriment Predictions* $C_{f_{n_f}}$ where $n_f = 3$: Minimum requirement for trajectory of quadratic functions must have at least 3 tuna nutriment predictions collected. To produce C_{f_3} , (9) is applied using $n_f = 3$ as parameter. Then, updating upper limit trajectory $\mathbf{T}_f^u(n_f)$ is denoted as follows:

$$\mathbf{T}_f(n_f) = \mathbf{T}_f^u(n_f), \quad (12)$$

5) *Tuna Nutriment Predictions* $C_{f_{n_f}}$ where $n_f \geq 4$: To precise accuracy of trajectory \mathbf{T}_f , we refine its trajectory by collecting more tuna nutriment prediction $C_{f_{n_f}}$. Tuna nutriment prediction $C_{f_{n_f}}$ is calculated using the nearest nutriment detection in area of $y^u = a_3^u x^2 + a_2^u x + a_1^u$ with tolerance degree from quadratic function between ± 30 degree.

To handle losing tuna nutriment prediction, we used previously tuna nutriment prediction by calculating the speed of nutriment in next frame.

$$\begin{aligned} \hat{x}_{f_{n_f}}^c &= 3\hat{x}_{f_{n_f-1}}^c - 3\hat{x}_{f_{n_f-2}}^c + \hat{x}_{f_{n_f-3}}^c, \\ \hat{y}_{f_{n_f}}^c &= a_3^c (\hat{x}_{f_{n_f}}^c)^2 + a_2^c \hat{x}_{f_{n_f}}^c + a_1^c, \end{aligned} \quad (13)$$

where a_1^c , a_2^c , and a_3^c are coefficients of quadratic function formed by trajectory \mathbf{T}_f

III. EXPERIMENT

In this section, we first explain the details of our datasets. We then describe evaluation approach to calculate error rate distance and show quantitative evaluation with various of n_f to discover an optimal value.

A. Datasets

We report our datasets containing 1 video which has interference of hand-held camera and ocean waves with 419 frames. Each dimension of frame has 1920×1080 pixels. Its video sequences are in MOV format with frame rate 30 frames/second. Range size of nutriment is starting from 9×6 to 13×36 pixels.

B. Evaluation Approach

Evaluation approach is defined by measuring minimum euclidean distance based on number of nutriment collected with ground truth \mathbf{T}_g . Best trajectory \mathbf{T}^* with minimum error rate distance is defined as:

$$\mathbf{T}^* = \arg \min_{n_f} (\mathbf{T}_g - \mathbf{T}(n_f))^T (\mathbf{T}_g - \mathbf{T}(n_f)), \quad (14)$$

where $n_f \in [3, 9]$.

C. Quantitative Evaluation with various of $n_f = [3, 9]$

Quantitative evaluation is computed by performance of detected nutriment and precision trajectory showed in Fig. 4. Number of detected nutriment is defined as percentage of detected nutriment divided by ground truth of nutriments appearing in frame. Meanwhile, precision trajectory is computed by total number of nutriments having trajectory leading to

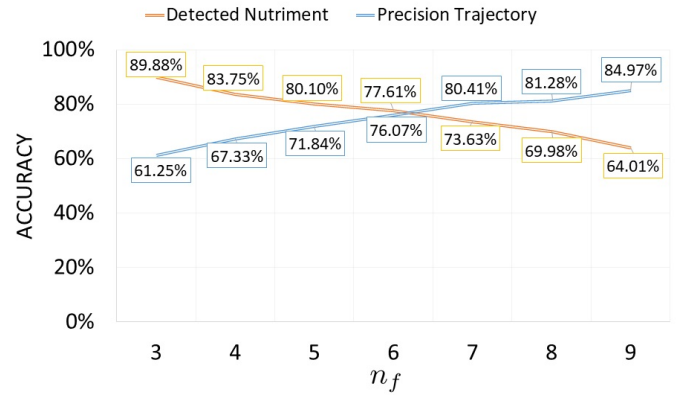


Fig. 4: Accuracy detected nutriment and precision trajectory in various $n_f \in [3, 9]$. The optimum value of detected nutriment and precision trajectory is $n_f = 6$.

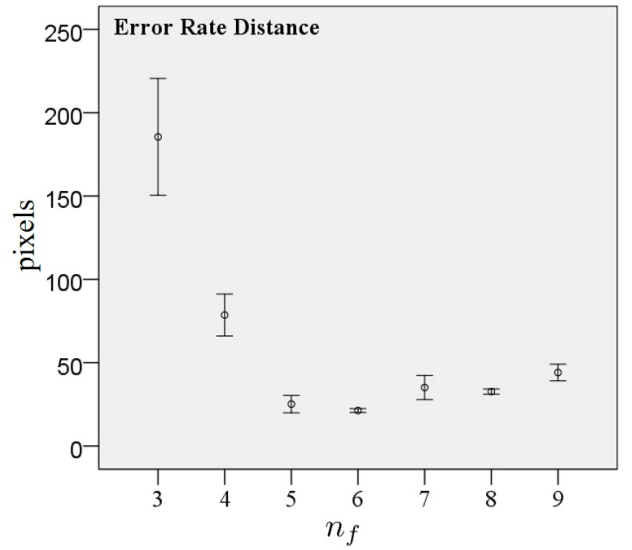


Fig. 5: 95% Confidence interval of error rate distance in various of $n_f \in [3, 9]$. The lowest error rate distance of proposed method is $n_f = 6$.

ripple area divided by detected nutriment. Fig. 5 and Table I show the confidence interval and statistical analysis of error rate distance in various of n_f . The results show that the optimal value of n_f is 6 in which this parameter produces smallest error rate distance.

IV. RESULT

In this section, we compare proposed method and state-of-the-art benchmark methods on our datasets. After that, we show the figures to explain the advantage of the proposed method and computational time between proposed method and state-of-the-art benchmark methods.

A. Evaluation Result

Precision of mAP in object detection is computed by performance YOLOv3 [36] to train our datasets with 10k

TABLE I: Statistical analysis of various $n_f \in [3, 9]$ for quantitative evaluation using one-samples T-Test.

nf	n	Mean (pixels)	Std. Dev. (pixels)	Std. Error (pixels)	95% Confidence Interval of the Diff.	
					Lower (pixels)	Upper (pixel)
3	30	185.47	93.81	17.13	150.44	220.5
4	30	78.58	33.64	6.14	66.02	91.14
5	30	25.17	13.97	2.55	19.96	30.39
6	30	21.32	3.08	0.56	20.18	22.48
7	30	35.12	19.34	3.53	27.9	42.35
8	30	32.67	4.19	0.76	31.11	34.24
9	30	44.1	13.32	2.43	39.12	49.07

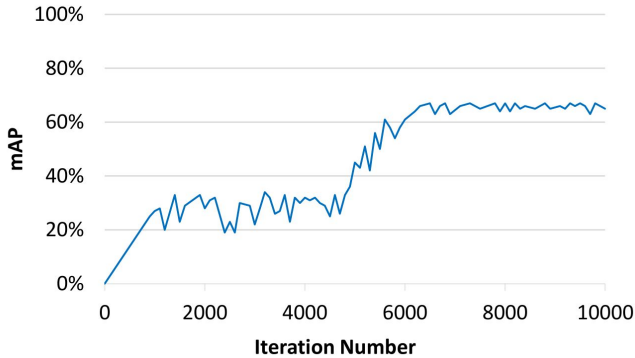


Fig. 6: mAP result during training model using YOLOv3 [36]. After 7k iterations, curve of mAP of model has more stable.

iterations with 416×416 pixels for image resizing from 1920×1080 pixels. Fig. 6 displays training result of our datasets using YOLOv3 [36] and reaches 67% of maximum mAP with 10k iterations. We also tested proposed methods and state-of-the-art benchmark results. There are many state-of-the-art methods using multiple object tracking (MOT) [38]–[44]. These methods perform well using six publicly available datasets on pedestrian detection, MOT and person search provided by [45]–[47]. In evaluations, we choose JDE [38] to represent MOT as benchmark method because JDE is very fast and accurate based on re-implementation of faster object detection compared with [39]–[44]. We also use SORT [48] as benchmark methods and add our detection model to completely understand performance of tracking method.

In Fig. 7a, the proposed method is demonstrated to be able to track small nutriment while JDE and SORT with original YOLOv3 and our detection model perform poor (Fig. 7b, 7c, 7d) without tracking results of nutriments even SORT is able to detect some nutriments. By our experiment, the benchmark methods fail to run our datasets because the size of nutriment is too small (maximum size is 13×36 pixels) and the speed of nutriment is fast (average nutriment movement from start to end node is 23.8 frames).

B. Implementation Details

For analysis of the computational complexity and execution time of the proposed methodology, a computational

TABLE II: Hardware and software environment for running proposed method and benchmark methods for comparison.

Spesification		
Hardware	CPU	Intel Core i7-9700 CPU @3.00GHz (8 CPUs)
	RAM	16 GB
	GPU	NVIDIA GeForce GTX 745
Software	OS	Windows 10 Pro 64-bit
	IDE	Microsoft Visual Studio Professional 2017 v.15.9.25
	Language	Python 3.6 64bit

TABLE III: Comparison computational time proposed method and benchmark methods.

Methods	N	Mean (fps)	Std. Dev. (fps)	Std. Err. (fps)	95% Confidence Interval of the Diff.	
					Lower (fps)	Upper (fps)
Ours	419	1.93	0.61	0.03	1.87	1.99
JDE	419	1.87	0.07	0.00	1.86	1.87
YOLOv3 + Sort	419	0.45	-	-	-	-
Our Detection + Sort	419	0.47	-	-	-	-

time analysis is conducted using a video with 419 frames. Table II shows the specification of hardware and software for comparison. Table III compares the computation time (in fps) for proposed method, namely trajectory mapping and benchmark approaches: JDE and SORT with original YOLOv3 and our detection model. For average and standard deviation of computational time, we reach 1.93 and 0.61 fps, while JDE spends 1.87 and 0.07 fps, respectively. SORT only provides average computational time without information of computational time for individual frame. Computational time for both detection model of YOLOv3 and our detection model with SORT performs worst and these benchmark approaches reach 0.45 fps and 0.47, respectively. By analyzing computational complexity, proposed method runs faster than JDE with the different speed is 0.6 fps.

V. CONCLUSION AND DISCUSSION

Tracking approach is the one of features to analyze fish behavior to create a decision to optimize the amount of nutriment. Recent studies have shown that it is possible to track movement objects in entire of frames on video. However, there is no agreement to track multiple small nutriments in the video which has interference of hand-held camera and ocean waves. In this paper, tuna nutriment tracking using trajectory mapping in application to aquaculture fish tank has been presented and demonstrated to be promising for interference video containing multiple small nutriment datasets. We have demonstrated tuna nutriment tracking using trajectory mapping and the method consistently performs well on the interference video with good precision trajectory result. We expect our approach to open the door for future work and to go beyond for feature extraction of ripple activity and focus on integrating tracking approach and ripple activity to be a decision to control fish feeding machine.



(a) Trajectory Mapping



(b) JDE



(c) YOLOv3+SORT



(d) Our Detection Model + SORT

Fig. 7: Observation of proposed method namely trajectory mapping shown in (a) and benchmark method results shown in (b), (c), and (d). Left and right images for each method represent first node of nutriment and nutriment tracked after several frames, respectively. In trajectory mapping method, red curve is defined as trajectory result of proposed method. Red box in each image represents ground truth of nutriment. We can see that both red curve and red box in trajectory mapping show precisely tracked result and it proves that trajectory mapping creates trajectory very well while benchmark methods perform poor without tracking results of nutriments even SORT is able to detect some nutriments.

REFERENCES

- [1] F. Fazio, *Fish hematology analysis as an important tool of aquaculture: A review*. *Aquaculture*, vol. 500, 237-242, 2019. doi:org/10.1016/j.aquaculture.2018.10.030
- [2] J. Freitas *et al.*, *From aquaculture production to consumption: Freshness, safety, traceability and authentication, the four pillars of quality*. *Aquaculture*, vol. 518, 734857, 2019. doi:org/10.1016/j.aquaculture.2019.734857.
- [3] B. Carmen *et al.*, *Seagrass meadows improve inflowing water quality in aquaculture ponds*. *Aquaculture*, vol. 528, 735502, 2020. doi:org/10.1016/j.aquaculture.2020.735502.
- [4] W. Liu *et al.*, *Characterizing the water quality and microbial communities in different zones of a recirculating aquaculture system using biofloc biofilters*. *Aquaculture*, vol. 529, 735624, 2020. doi:org/10.1016/j.aquaculture.2020.735624.
- [5] U. Farheen *et al.*, *Automatic Controlling of Fish Feeding System*. *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, vol. 6, Issue. 7, 362-367, 2018.
- [6] H. Liu *et al.*, *Biofloc formation improves water quality and fish yield in a freshwater pond aquaculture system*. *Aquaculture*, vol. 506, 735624, 2019. doi:org/10.1016/j.aquaculture.2019.03.031.
- [7] K. Higuchi *et al.*, *Effect of long-term food restriction on reproductive performances in female yellowtail, *Seriola quinqueradiata**. *Aquaculture*, vol. 486, 224 - 231, 2018. doi:org/10.1016/j.aquaculture.2017.12.032.
- [8] J.M. Barron *et al.*, *Evaluation of effluent waste water from salmonid culture as a potential food and water supply for culturing larval Pacific lamprey *Entosphenus tridentatus**. *Aquaculture*, vol. 517, 734791, 2020. doi:org/10.1016/j.aquaculture.2019.734791.
- [9] Y. Atoum *et al.*, *Automatic Feeding Control for Dense Aquaculture Fish Tanks*. *IEEE Signal Process. Lett.*, vol. 22, 1089-1093, 2015. doi:10.1109/LSP.2014.2385794, 2015.
- [10] A.K. Sabari *et al.*, *Smart Fish Feeder*. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 2, Issue. 2, 111-115, 2017.
- [11] P.C. Oostlander *et al.*, *Microalgae production cost in aquaculture hatcheries*. *Aquaculture*, vol. 525, 735310, 2020. doi:org/10.1016/j.aquaculture.2020.735310.
- [12] C. J. Bridger and R. K. Booth, *The effects of biotelemetry transmitter presence and attachment procedures on fish physiology and behavior*. *Rev. Fisheries Sci.*, vol. 11, No. 1, 13-34, 2003.
- [13] S. G. Conti *et al.*, *Acoustical monitoring of fish density, behavior, and growth rate in a tank*. *Aquacult. Eng.*, vol. 251, No. 2, 314-323, 2006.
- [14] C. Costa *et al.*, *Extracting fish size using dual underwater cameras*. *Aquacult. Eng.*, vol. 35, No. 3, 218-227, 2006.
- [15] J. Xu *et al.*, *Behavioral responses of tilapia (*Oreochromis niloticus*) to acute fluctuations in dissolved oxygen levels as monitored by computer vision*. *Aquacult. Eng.*, vol. 35, No. 3, 207-217, 2006.
- [16] L.H. Stien *et al.*, *A video analysis procedure for assessing vertical fish distribution in aquaculture tanks*. *Aquacult. Eng.*, vol. 37, No. 2, 115-124, 2007.
- [17] B. Zion *et al.*, *Real-time underwater sorting of edible fish species*. *Comput. Electron. Agricult.*, vol. 56, No. 1, 34-45, 2007.
- [18] S. Duarte *et al.*, *Measurement of sole activity by digital image analysis*. *Aquacult. Eng.*, vol. 41, No. 1, 22-27, 2009.
- [19] Y. Atoum *et al.*, *Automatic Feeding Control for Dense Aquaculture Fish Tanks*. *IEEE Signal Processing Letters*, vol. 22, No. 8, 1089-1093, 2015, doi: 10.1109/LSP.2014.2385794.
- [20] M.A. Adegboye *et al.*, *Incorporating Intelligence in Fish Feeding System for Dispensing Feed Based on Fish Feeding Intensity*. *IEEE Access*, vol. 8, 91948-91960, 2020, doi: 10.1109/ACCESS.2020.2994442.
- [21] A. Ess *et al.*, *A mobile vision system for robust multi-person tracking*. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1-8, 2008.
- [22] M. Breitenste *et al.*, *Robust tracking-by-detection using a detector confidence particle filter*. *IEEE International Conference on Computer Vision (ICCV)*, 1515-1522, 2009.
- [23] S. Pellegrini *et al.*, *You'll never walk alone: modeling social behavior for multi-target tracking*. *IEEE International Conference on Computer Vision (ICCV)*, 261-268, 2009.
- [24] T. Ueno *et al.*, *Motion-blur-free microscopic video shooting based on frame-by-frame intermittent tracking*. *IEEE International Conference on Automation Science and Engineering (CASE)*, 837-842, 2015, doi: 10.1109/CoASE.2015.7294185.
- [25] J. Berclaz *et al.*, *Robust people tracking with global trajectory optimization*. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 35, No. 3, 744-750, 2006.
- [26] X. Zhang and E. Izquierdo, *Real-Time Multi-Target Multi-Camera Tracking with Spatial-Temporal Information*. *IEEE Visual Communications and Image Processing (VCIP)*, 1-4, 2019, doi: 10.1109/VCIP47243.2019.8965845.
- [27] J. Berclaz *et al.*, *Multiple object tracking using k-shortest paths optimization*. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 33, No. 9, 1806-1819, 2011.
- [28] H. Jiang *et al.*, *A linear programming approach for multiple object tracking*. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1-8, 2007.
- [29] L. Zhang *et al.*, *Global data association for multi-object tracking using network flows*. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1-8, 2008.
- [30] H. Pirsiavash *et al.*, *Globally optimal greedy algorithms for tracking a variable number of objects*. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1201-1208, 2011.
- [31] L. Leal-Taixe *et al.*, *Learning an image-based motion context for multiple people tracking*. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [32] L. Leal-Taixe *et al.*, *Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker*. *IEEE International Conference on Computer Vision (ICCV) Workshops. 1st Workshop on Modeling, Simulation and Visual Analysis of Large Crowds*, 2011.
- [33] A. Zamir *et al.*, *Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs*. *ECCV*, 2012.
- [34] S. Tang *et al.*, *Multi people tracking with lifted multicut and person re-identification*. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [35] Nghia, *Simple video stabilization using OpenCV*. <http://nghiaho.com/?p=2093>, 2010.
- [36] J. Redmon and A. Farhadi, *YOLOv3: An Incremental Improvement*. *CoRR*, vol. abs/1804.02767, 2018.
- [37] Weisstein and W. Eric, *Least Squares Fitting-Polynomial*. From *MathWorld-A Wolfram Web Resource*, <https://mathworld.wolfram.com/LeastSquaresFittingPolynomial.html>.
- [38] Z. Wang *et al.*, *Towards Real-Time Multi-Object Tracking*. *CoRR*, vol. abs/1909.12605, 2019, <http://arxiv.org/abs/1909.12605>.
- [39] W. Lin *et al.*, *Real-time multi-object tracking with hyper-plane matching*. Technical report, , Shanghai Jiao Tong University and ZTE Corp, 2017.
- [40] S. Tang *et al.*, *Multiple People Tracking by Lifted Multicut and Person Re-identification*. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 2017, 3701-3710, 2017, doi: 10.1109/CVPR.2017.394.
- [41] Y. Zhang *et al.*, *A Simple Baseline for Multi-Object Tracking*. *CoRR*, vol. abs/2004.01888, 2020, <https://arxiv.org/abs/2004.01888>.
- [42] F. Yu *et al.*, *POI: Multiple Object Tracking with High Performance Detection and Appearance Feature*. *Computer Vision - ECCV 2016 Workshops - Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II*, vol. 9914, 36-42, 2016, 10.1007/978-3-319-48881-3_3.
- [43] M. Babae *et al.*, *A dual CNN-RNN for multiple people tracking*. *Neurocomputing*, vol. 368, 69-83, 2019, 10.1016/j.neucom.2019.08.008.
- [44] B. Pang *et al.*, *TubeTK: Adopting Tubes to Track Multi-Object in a One-Step Training Model*. *CoRR*, vol. abs/2006.05683, 2020, <https://arxiv.org/abs/2006.05683>.
- [45] A. Milan *et al.*, *MOT16: A Benchmark for Multi-Object Tracking*. *CoRR*, vol. abs/1603.00831, 2016, <http://arxiv.org/abs/1603.00831>.
- [46] L. Leal-Taixe *et al.*, *MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking*. *CoRR*, vol. abs/1504.01942, 2015, <http://arxiv.org/abs/1504.01942>.
- [47] A. Ess *et al.*, *Depth and Appearance for Mobile Scene Analysis*. *IEEE 11th International Conference on Computer Vision, ICCV 2007, Rio de Janeiro, Brazil, October 14-20, 2007, 1-8, 2007*, doi:org/10.1109/ICCV.2007.4409092.
- [48] Bewley *et al.*, *Simple online and realtime tracking*. *2016 IEEE International Conference on Image Processing (ICIP)*, 3464-3468, 2016, 10.1109/ICIP.2016.7533003.