Intelligent Underwater Stereo Camera Design for Fish Metric Estimation Using Reliable Object Matching

Naomi A. Ubina¹, ¹, Shyi-Chyi Cheng¹, Member, IEEE, Chin-Chun Chang¹, Member, IEEE, Sin-Yi Cai¹, Hsun-Yu Lan², and Hoang-Yang Lu³

¹Department of Computer Science and Engineering, National Taiwan Ocean University, Keelung City, Taiwan
²Department of Aquaculture, National Taiwan Ocean University, Keelung City, Taiwan
³Department of Electrical Engineering, National Taiwan Ocean University, Keelung City, Taiwan
⁴College of Computing Studies, Information and Communications Technology, Isabela State University, Isabela, Philippines

Corresponding author: Shyi-Chyi Cheng (e-mail: csc@mail.ntou.edu.tw).

This work was supported in part by the Ministry of Science and Technology and Fisheries Agency, Council of Agriculture, Taiwan, under grant numbers MOST 110-2221-E-019-048- and 111AS-6.2.1-FA-F6, respectively.

ABSTRACT The resolution of the computed depth maps of fish in an underwater environment limits the 3D fish metric estimation. This paper addresses this problem using object-based matching for underwater fish tracking and depth computing using convolutional neural networks (CNNs). First, for each frame in a stereo video, a joint object classification and semantic segmentation CNN is used to segment fish objects from the background. Next, the fish objects in these images are cropped and matched for the subpixel disparity computation using the video interpolation CNN. The calculated disparities and depth values estimate the fish metrics, including the length, height, and weight. Next, we tracked the fish across frames of the input stereo video to compute the metrics of the fish frame by frame. Finally, the fish median metrics are calculated for error rate in our fish length estimation. Experimental results show a 5% error rate in our fish length estimation.

INDEX TERMS convolutional neural network, object tracking, object-based stereo matching

I. INTRODUCTION

Deep learning methods for stereo image processing are gaining popularity. Many studies have been involved in stereo imaging [1][2][3], but only a few applications have been applied to modeling underwater scenes. Underwater images have more significant challenges, as images that were taken generally have problems with image degradation, poor contrast, blurring, color deviation [4], poor visibility, light attenuation, and water turbidity [5]. For stereo images, depth information and disparity map information are essential for quality 3D modeling [6][7][8]. To integrate an underwater stereo camera for 3D modeling in the underwater environment, we used it to capture and match each fish using the left and right images based on the proposed uncompressed underwater stereo matching neural network. The dense disparities obtained between the left and right fishes are computed to obtain the depth map of the 3D model of each fish. Through this 3D model, the fish's body length, height, and width can be estimated more accurately. Estimated values combined with various sensors and weight regression formulas can establish the growth curve of the fish. Monitoring the size of the fish can help breeding experts record the breeding process and reduce labor costs.

Figure 1 shows the diagram of our proposed stereo matching for underwater object reconstruction using the left and right images as the inputs. The stereo image rectification is a preprocessing technique to obtain the correct intrinsic and extrinsic parameters by calibrating the stereo camera system. Then, each corrected image is inputted to the...
instance segmentation neural network to transform each image frame into a set of fish objects and background objects. For object matching, the correspondence in the left image is searched in the right image to generate the disparity map. But a single disparity value cannot accurately restore the pixel depths of the 3D object.

Another potential problem is that the object’s pose in the left image may be slightly different from that of the right image because the left and right cameras of the low-cost stereo camera system might not be accurately synchronized. Also, the target fish object has multiple gestures since it freely swims in the underwater environment, which brings additional noise in measuring the exact 3D information of the fish. Lastly, the fish might overlap with other fish in the captured images, which degrades the mask accuracy of the fish objects even with well-designed instance segmentation CNNs.

In solving these mentioned difficulties, the left image object and the right image object are cropped and aligned to form the input pair, which is further processed. Our video interpolation CNN (VICNN) [9] based stereo matching algorithm calculates the residual disparity of each pixel in the left object. The core of our stereo matching algorithm is VICNN, which synthesizes the intermediate object to establish the pixel correspondences between the left and right objects. Instead of using a single frame image, based on the proposed object matching scheme, we tracked each fish across frames and calculated a sequence of 3D models to reduce the biometric noise introduced by the gesture variations of a freely swimming fish. This mechanism is nonintrusive and reduces manual handling of the fish to prevent stress [10] and disturbance and avoids injury caused by fish catching in estimating the biological information.

The contributions of the proposed approach are as follows. First, the proposed deep neural network establishes real-time pixel correspondences between stereo images. Second, traditional deep learning models require a large set of human-annotated label data for training. Our approach is directly trained from the raw video data; thus, the deep neural networks training complexity for 3D model reconstruction is significantly reduced. Next, it is difficult to establish precise pixel correspondences from the texture-less stereo images using traditional stereo matching algorithms. The interpolated signals of the matched object pairs ensure the correctness of the computed disparity image from precise correspondences with minimal object matching error. Finally, the object-based stereo matching optimization algorithm contributes to the design of the image warping of the disparity to minimize the resulting smoothness. The remainder of this paper is as follows. Section 2 provides the details of our methods, and Section 3 is on the experimental results. Finally, Section 4 is our conclusions and future works.

II. MATERIALS AND METHODS

A. RELIABLE OBJECT MATCHING WITH SEMANTIC SEGMENTATION CNN

As shown in Figure 1, our approach integrates camera calibration for preprocessing as the first stage to obtain the stereo system’s correct intrinsic and extrinsic parameters. Then, the internal and external parameters are inputted into the image rectification process to obtain the rectified images. This rectification process projects images using a common horizontal image plane as it twists back the left and right image pixels to have the same coordinates in the horizontal plane. Image rectification yields all epipolar lines to be parallel in the image plane. For illustration, we assumed that the input stereo images were rectified before being applied to the proposed system for further processing.

FIGURE 1. Block diagram of our stereo matching-based underwater scene modeling.

The capturing theory of the stereo camera system: (a) an example of the captured stereo images; (b) the theory to compute the disparity and depth of the object. The y-axis is perpendicular to the page.

Each rectified left, and right image pair is first inputted to a semantic segmentation neural network [11] with excellent object detection and segmentation performance. The semantic segmentation CNN transforms each frame image into a set of reliable fish objects and the background object. Let \( O_t^{L} \) (\( O_t^{R} \)) and \( O_{t+1}^{L} \) (\( O_{t+1}^{R} \)) be the tracked object in frames \( t \) and \( t+1 \) of the left (right) video, respectively. The object pair \( (O_t^{L}, O_t^{R}) \) is a stereo object if the matched cost
between $O_t^L$ and $O_{t+1}^L$ is small, and their motion vectors are similar. Given the object $O_t^L$, the basic processing of object matching is to search the corresponding object $O_t^R$ in the right image along the x-axis since the images have been rectified to have a horizontal epipolar geometry. Figure 2 depicts the basic concept of image capturing using a stereo camera system. In Figure 2, the foreground object, i.e., the fish, is overlapped with different backgrounds in the rectified left and right images, which decreases the accuracy of object matching. Thus, we perform the semantic segmentation to determine, the matching cost defined in (2) for stereo matching. Thus, we perform the semantic segmentation CNN to obtain the masks of $O_t^L$ and $O_t^R$, in which the backgrounds are removed.

The object matching scheme first computes the motion vector between $O_t^L$ ($O_t^R$) and $O_{t+1}^L$ ($O_{t+1}^R$) based on the computation of the matching cost with the object pair ($O_t^L$, $O_{t+1}^L$) ($O_t^R$, $O_{t+1}^R$) as the input. The matching cost between objects $O_t$ and $O_{t+1}$ is measured by aggregating pixel-wise matching costs in both objects with the support weights [12]. Let $x_t$ ($x_{t+1}$) be the center of the object $O_t$ ($O_{t+1}$). The support weight between pixels $x_t$ and $x_{t+1}$ is defined as

$$w(x_t, x_{t+1}) = \exp\left(-\left(\frac{\Delta c_{x_t,x_{t+1}}}{\sigma_c} + \frac{\Delta g_{x_t,x_{t+1}}}{\sigma_g}\right)\right)$$  \hspace{1cm}  (1)

where $\Delta c_{x_t,x_{t+1}}$ and $\Delta g_{x_t,x_{t+1}}$ represents the color difference and the spatial distance between pixels $x_t$ and $x_{t+1}$, respectively; $\sigma_c$ is the variance of color difference; $\sigma_g$ is determined according to the size variance of all the objects. The value of $w(x_t, x_{t+1})$ measures the strength of the pixel correspondence ($x_t$, $x_{t+1}$). Notice that the motion vector of the pixel $x_t$ can be computed as $u_{x_t} = x_{t+1} - x_t$. To assume every pixel in $O_t$ would have a similar motion vector, we can compute the matching cost of the object pair ($O_t$, $O_{t+1}$) by combining the pixel-wise support-weights in both objects:

$$E(x_t, x_{t+1}) = \frac{\sum_{p \in O_{t+1} \cap O_t} w(x_t, p) w(x_{t+1}, p)}{\sum_{p \in O_t} w(x_t, p) w(x_{t+1}, p)} E_{p\in O_{t+1}}(x_t, x_{t+1})$$  \hspace{1cm}  (2)

where the pixel pair $(p_t, p_{t+1})$ is constrained to have the motion vector $u_{x_t}$. Using (2), for each object $O_t$ in frame $t$, we can define the matched object $O_{t+1}'$ with the center at $x_{t+1}'$ in frame $t+1$ as

$$O_{t+1}' = \arg \min_{O_{t+1} \in S_{t+1}(x_t)} E(x_t, x_{t+1})$$  \hspace{1cm}  (3)

where $S_{t+1}(x_t)$ is the set of all possible objects within the search window with $x_t$ as the center in frame $t+1$.

Once the motion vector $u_t^L$ ($u_t^R$) of the object $O_t^L$ ($O_t^R$) is determined, the matching cost defined in (2) for stereo object searching can be refined as

$$E_S(x_t^L, x_t^R) = E(x_t^L, x_t^R) + \lambda \|u_t^L - u_t^R\|_2$$  \hspace{1cm}  (4)

for each object $O_t^L$ in the left image, we can define the best-matched object $O_t^{R*}$ with the center at $(x_t^{R*}, y)$ in the right image as

$$O_t^{R*} = \arg \min_{O_t^R} E_S(x_t^L, x_t^R)$$  \hspace{1cm}  (5)

where $S_0$ is the set of all possible objects in the right image for matching the object $O_t^L$. Notice the disparity of object $O_t^L$ with the center pixel $x_t^L$ can be computed as $d_t = x_t^L - x_t^{R*}$.

**B. PIXEL-WISE RESIDUAL DISPARITY ESTIMATION WITH VIDEO INTERPOLATION CNN**

![Figure 3. Establishing the stereo matching algorithm for residual disparity computation based on the video interpolated CNN: (a) the underwater stereo imaging system; (b) the schematic diagram of using the stereo imaging system to monitor underwater fish; (c) given a pair of rectified stereo underwater images $(O_t^L, O_t^R)$, the video interpolation CNN (VICNN) [9] produces a set of pixel-wise kernels $K_L$ and $K_R$ for synthesizing the middle object $O_{t+1/2}$ and the displacement image $d_t^L$.](image)

Obviously, the pixel-wise motion vectors (optical flow) in an object are similar but not the same because individual parts of the object might perform different actions. Similarly, the pixel-wise disparities of an object are not the same as that of center pixel since the depth information of a real-world 3D object is not the same everywhere. As mentioned above, given a detected object pair $(O_t^L, O_t^R)$, Equation (5) can compute the basic disparity for each pixel in $O_t^L$. Obviously, for each pixel $x$ in $O_t^L$, this disparity difference $\Delta d_x$ should be estimated to model the disparity of $x$ as

$$d_t = d_t^L + \Delta d_x$$  \hspace{1cm}  (6)

where $d_t^L$ is the disparity value of the center pixel $x_t^L$ defined by Eq. (5). Suppose we translate the centers of matched objects into the common original point $(0,0)$. In that case, the left object and the right object are aligned with each other and form a new object pair $(O_t^L, O_t^R)$ which can be used to estimate the pixel disparity difference $\Delta d_x$ of the pixel $x$ in $O_t^L$.

Figure 3 shows the proposed stereo matching algorithm based on the video interpolated CNN (VICNN) [9], synthesizing the middle object $O_{t+1/2}$ with the object pair $(O_t^L, O_t^R)$ as the input. For each pixel $x$ in $O_{t+1/2}$, the VICNN computes a pixel-wise kernel pair $(K_L(x), K_R(x))$ to...
interpolate the pixel value of \( x = (x, y) \) in \( O_{t,R}^i \) using the following equation:

\[
f(x) = < K_L(x), P_L(x) > + < K_R(x), P_R(x) >
\]  

(7)

where \( < \cdot, \cdot > \) is the inner product operator; \( P_L(x) \in O_{t,L}^i \) and \( P_R(x) \in O_{t,R}^i \) are patches with the common center \( x \). The kernel pair can also be used to compute the disparity difference of \( x \):

\[
\Delta d_x = < K_R(x), U_x(x) > - < K_L(x), U_x(x) >
\]  

(8)

where \( U_x \) is the \( x \)-displacement matrix with \( U_x(x') = x' - x \). Eq. (8) defines the displacement image as

\[
D_{x}^{t-R} = \{ \Delta d_x \}_{x \in O_{t}^i}.
\]  

(9)

Although CNN-based video interpolation can generate accurate interpolated images for both uniform regions and edges, it cannot ensure the correctness of displacement vectors for pixels in uniform regions. Therefore, instead of proposing a new architecture for VICNN, we modified the loss function used by VICNN by adding additional metrics to improve or optimize and further re-train VICNN using our own set of underwater training videos to precisely generate the displacement image \( D_{x}^{t-R} \) for the underwater image pair \( (O_{t,L}^i, O_{t,R}^i) \).

We revised the training procedure of the original VICNN by integrating the total variation of the detected displacement vectors to ensure that the estimated displacement vectors will be very smooth. The authors of VICNN [9] used two input receptive patches \( R_{t,1} R_{t,2} \) at the center of \( (x_t, y_t) \) with the corresponding input patches \( P_{t,2} P_{t,2} \) which are smaller than the receptive field patches where both are also centered with the same location. \( \tilde{G}_t \) is the ground truth color and \( \tilde{G}_t \) is the ground truth gradient at \( (x_t, y_t) \). Initially, the loss function measures the difference between the interpolated pixel color and its corresponding ground truth defined as:

\[
E_c = \sum L_1 \| [P_{t,1} P_{t,2}] * K_t - \tilde{G}_t \|_{L_1}
\]  

(10)

where subscript \( i \) is the \( i^{th} \) training example and \( K_t \) is the output of the neural network’s convolutional kernel. Using only the color loss and even with the integration of \( L_1 \) norm, which is the sum of the absolute values of the distances in the original space to preserve the edges of the image [13], still leads to a blurry result. The integration of gradients in the loss function corrects the shortcoming of the color loss. The gradients of the input patches are first computed followed by convolution using the estimated kernel, which generates the gradient of the interpolated image at the pixel interest. Based on the eight immediate neighboring pixels, eight versions of gradients were computed using finite difference and all added into the gradient loss function defined as:

\[
E_g = \sum_{k} \sum_{i=1}^{B} \| G_{t,k}^{i} G_{t,2}^{i} * K_t - \tilde{G}_t \|_{L_1}
\]  

(11)

where \( k \) belongs to the eight ways to compute the gradient. Meanwhile, \( G_{t,k}^{i} \) and \( G_{t,2}^{i} \) are the gradients based on input patches \( P_{t,2} P_{t,2} \) and the input ground-truth gradient is in \( \tilde{G}_t \). The final loss of VICNN combines the color and gradient loss as

\[
E_f = E_c + \lambda \cdot E_g
\]  

(12)

where \( (\lambda = 1) \) determines the smoothness of the output. To verify the quality of the displacement vector estimation of the original loss function \( E_f \) of VICNN, we also add TVL1 [14] factor as an additional term of (12), a popular approach to remove impulse noise and preserve image edges [13] to optimize the optical flow detection. We integrated the sum of the gradient values of the displacement vectors, i.e., \( |\nabla D_{x}^{t-R}| \) as the total variation function for \( x, y \) to smoothen the results of the detected displacement vectors. The final loss function \( E_f \) is denoted as

\[
E_f = (E_c + \lambda \cdot E_g) + (|\nabla U_{t,x}| + |\nabla U_{t,y}|).
\]  

(13)

The gradient provides shape information and TVL1 for temporal information and will increase its displacement vector estimation’s overall robustness and reliability.

In training the neural network, image ground truth is needed to train the parameters of the neural network. We follow the concept of VICNN using three consecutive video frames of both the left \( (I_{t,1}, I_{t+1}, I_{t+2}) \) and right \( (I_{t,1}, I_{t+1}, I_{t+2}) \) incorporating the second frame \( (I_{t+1}) \) as our ground truth and will help determine the final parameters of VICNN to generate the displacement vectors using (8). Figure 4 shows an example to estimate the final disparity image of the target based on the object-based stereo matching algorithm.

FIGURE 4. An example of the underwater fish object disparity estimation using the proposed approach: (a) the original stereo images; (b) the rectified stereo images; (c) the interpolated image of (b); (d) the computed disparity image of the fish object in (c).
Once the disparity value of pixel \( i \) in \( O^L_i \) has been computed, the depth value of the pixel and its 3D coordinates can be computed as:

\[
[x, y, z] = [(x_i - c_x) \cdot Z_i / f, (y_i - c_y) \cdot Z_i / f, f \cdot b / d_i] \tag{14}
\]

where \( f \) is the camera’s focal length; \( b \) is the baseline, which is defined as the distance between the centers of the left and right cameras; \((c_x, c_y)\) is the optical center in the image plane; \( d_i \) is the computed disparity value of pixel \( i \). For each pixel in the object \( O^L_i \), we use Eq. (14) to compute its corresponding 3D point. This also defines the point cloud \( P^L_i = \{(X_i, Y_i, Z_i)\}_{i=1}^{\text{obj}} \) of the fish contained in \( O^L_i \). Then we perform Singular Values Decomposition (SVD) for the 3D point cloud \( P^L_i \) and the matrix can be disassembled to generate the following:

\[
P^L_i = U \Sigma V^T \tag{15}
\]

where \( \Sigma \) is a diagonal matrix composed of three eigenvalues \( \lambda_i = 1, ..., 3 \); the columns of \( U \) and \( V \) are called the left-singular vectors and right-singular vectors of \( P^L_i \), respectively. Without loss of the generality, we often have \( V = [v_1]_{i=1}^{3} \), where \( v_i \) is the \( i \)-th eigenvector of the matrix \( P^L_i \).

![Figure 5. The fish objects are labeled either 'side' or 'front' view for training our semantic segmentation CNN: (a) example of the 'side view' fish; (b) example of the 'front view' fish; (c) the body length \( L \) and body height \( H \) of (a); (d) body height \( H \) and body width \( W \) of (b); (e) segmented fish objects with correct posture labels from the input image in the testing phase.](image)

For each point \( p \in P^L_i \), we project \( p \) onto the three eigenvectors \( v_1, v_2, v_3 \) (defined by the matrix \( V \)) to get the new 3D coordinate points:

\[
[X_i, Y_i, Z_i] = [x_i^j, y_i^j, z_i^j], i = 1, ..., |O^L_i|. \tag{16}
\]

The converted coordinate points estimate the body length \( L \), body height \( H \), and body width \( W \) of the fish based on the fish posture recognition result. As shown in Figure 5, the training fish objects are labeled either ‘side view’ or ‘front view’ to train our semantic segmentation CNN to segment fish objects with correct posture labels from the input image in the testing phase. Based on the posture label and the converted coordinate points using (16), the formula to estimate the fish metrics is as follows:

\[
\begin{bmatrix}
L \\
H \\
W
\end{bmatrix} = \begin{bmatrix}
\max(x_i) - \min(x_i) \\
\max(y_i) - \min(y_i) \\
\max(z_i) - \min(z_i)
\end{bmatrix}, \text{if } O^L_i \text{ is 'side view'}. \tag{17}
\]

In this work, the objects identified as ‘side view’ class are considered the reliable ones for fish metric estimation, while the ‘front view’ objects are skipped to avoid having extra noises in the fish metric measurement results.

Another factor that affects the estimation accuracy of underwater fish metrics is the distance \( b \) between the left and right camera lenses. The larger the value of \( b \), the more accurate the fish metrics will be. However, increasing the distance requires a large stereo camera which is difficult to use in offshore cages. Thus, we used a small value of \( b \) \((b = 11.4 \, \text{cm})\) to set up the camera though it is limited in measuring the metrics of the fish far away from the camera.

With this set of cameras, we perform different experiments to determine which among them are effective ranges. First, we used a fake fish with a body length of 30cm with a distance of 70cm and 90cm from the lens. To test further, we also used shooting distances of 120cm, 150cm, 180cm, and 200cm, as shown in Figure 6. After calibrating the captured images, we used the method proposed in this paper to estimate the body length and use the error between the estimated body length and the actual body length for comparison. Finally, we compared the results of the estimated body length of the fish using our captured fake fish images using different distances to get the best result. Based on Table 1, the highest error rate (9.6%) is at a 200 cm distance and is considered the maximum effective distance range from the camera lens. Fish more than 200cm away from the camera lens are discarded in the body length estimation due to a higher error rate.

**TABLE 1**

| Distance between the lens and the fake fish | Fake fish body length estimation | Error (%) |
|-------------------------------------------|----------------------------------|-----------|
| 70 cm                                     | 30.7 cm                          | 2%        |
| 90 cm                                     | 30.5 cm                          | 1.6%      |
| 120 cm                                    | 30.2 cm                          | 0.6%      |
| 150 cm                                    | 30.6 cm                          | 2%        |
| 180 cm                                    | 31.6 cm                          | 5%        |
III. EXPERIMENTAL SET-UP AND DATASETS

We used two GoPro cameras to attain the stereo camera lens setup, as shown in Figure 3(a). The computer used for training the neural network for object detection, image matching, and 3D reconstruction is an Intel Core i7-8700 3.2GHz CPU, 32.0GB RAM, and NVIDIA GeForce GTX 1080ti GPU using the Python environment. The Pool 13 of the University Aquatic Center is located at the National Taiwan Ocean University in Keelung City, with 150 porphyry sculptures, and the Pingtung Hengchun Aquaculture site with approximately 40,000 golden pomfret was the experimental environment. Two environments were considered to ensure that our approach works for less dense and highly dense aquaculture tanks or cages. The video collected from these locations was used to train and test the neural network using the stereo camera to capture the left and right images. Figure 7(a) shows the labeled images for training our semantic segmentation CNN and the video interpolation CNN. A total of 2 hours, 25 minutes, and 3 seconds were taken from the A13 Aquatic Center Pond, while a total of 1 hour, 29 minutes, and 41 seconds from the Hengchun Ocean open-sea cage were used for the experiment.

FIGURE 6. Images of fake fish at distances 70, 90, 120, 150, 180, and 200 cm from the stereo camera lens.

IV. EXPERIMENTAL RESULTS

The left and right underwater images were taken by the stereo camera passed through the Mask_RCNN neural network, where the fish and the background are segmented through instance segmentation. There are two types of fish species, where the ponds contain porphyria sea bream and the Hengchun ocean has the golden pomfret. In training the Mask-RCNN neural network to perform instance segmentation, we manually labeled the images and utilized 200 images for the training data and 500 images for the testing data that were taken from the Aquatic Center with an accuracy rate of 90%. The segmentation results are shown in Figure 8(a). Meanwhile, from the data collected from the Hengchun aquaculture site, 500 images were used for the training and 800 images for the testing with an accuracy rate of 85%, and segmentation results are shown in Figure 8(b).

FIGURE 7. Stereo video datasets for training and testing: (a) images from the Aquatic Center; the upper row is the original image, and the bottom is the labeled data; (b) images captured from A13 Aquatic Center Pond; and (c) images captured from Hengchun Ocean open-sea cage.

FIGURE 8. Segmentation results: (a) A13 pool in the Aquatic Center and (b) from the Hengchun offshore fish cage.

The Mask-RCNN training and testing loss result is shown in Figure 9. As reflected in the loss curves for training and testing, the loss value close to zero is at the 200th iteration. Segmenting underwater target objects from their background is challenging considering fish's continuous or active movement, varied textures, quality of water, and luminosity. These problems should be considered more robust and accurate fish detection as a requirement for fish length and density estimation.
The detection accuracy results for the two data collection sites for Mask-RCNN are shown in Table 2. Accuracy has different results for the two environments since the quality of the collected video data varies in terms of water quality and turbidity and affects the identification of the target objects. Video collected from the pool environment with fewer fish populations has a higher accuracy rate of 95% compared to the open fish cage with a dense fish population of 90%.

TABLE 2

| Field Area          | Video Data | Accuracy |
|---------------------|------------|----------|
| Aquatic Center      | Video 1    | 92%      |
| A13 Pool            | Video 2    | 90%      |
|                     | Video 3    | 95%      |
|                     | Video 4    | 87%      |
| Hengchun Offshore   | Video 1    | 90%      |
| Fish Cage           | Video 2    | 84%      |
|                     | Video 3    | 88%      |
|                     | Video 4    | 87%      |

After rectification, the proposed object matching scheme tracks objects across video frames and searches the matched right object for each left object, whose initial disparity value is then determined by the displacement vector between the centers of the matched objects. The results of the pairing of the 3D object using the left and right images are shown in Figure 10. The matching accuracy for the images collected from the A13 pool of the Aquatic Center is 90%, while the Hengchun open-sea cage of the ocean is 80%. The lower result from Hengchun is due to the highly dense fish population, making it hard to perform matching due to high fish object overlaps.
Figure 13 shows the results for comparing the disparity of the stereo image matching using our approach with semi-global block matching (SGBM) [15] that integrates pixel-wise matching based on mutual information and the approximation of the global smoothness constraint. SGBM detects occlusions and disparities using subpixel accuracy. The results using our video interpolated optical flow, even with poor image quality due to water turbidity, have better disparity results with the results generated by SGBM. This is because the structure or appearance of the fish using our method is more visible and apparent. To calculate the disparity of pixel $x$ in the object $O_i$, the depth value of pixel $x$ and its 3D coordinates $X, Y, Z$ are 
\[ [X, Y, Z] = \{ (x - c_x) \ast Z / f, (y - c_y) \ast Z / f, f \ast b / d_x \} \]
where $f$ is the focal length of the camera and $b$ is the baseline or the distance between the center of the left lens and the center of the lens. The said formula is used to convert the 2D coordinate of the object $O_i$ into a 3D point cloud coordinate. To perform filtering for better results, we modified the outlier point cloud.

Based on our fake fish experiment results in Table 2, the most effective stereo camera lens distance is 200cm, and fish beyond that distance will obtain a significant error. Though the stereo camera system is an affordable device to get the image depth and can be used to estimate object size, it is only limited to a certain distance; such intervention was made to address its limitations and make sure to have the best optimization results. We also considered the continued movement of the fish, where their body and tails swing while swimming, which makes their estimated body length different for each frame. We used a tracking method to estimate the fish body length in each frame and calculate the average accumulated body lengths of different frames to deal with this. The obtained average value will be used as the final estimated body length of the fish.

To verify the error value with the actual body length of the fish, we track a single fish and measure its body length in full frames. We first manually measured the exact body length of the single fish, placed it separately in the water tank, and took a video using a stereo camera. We tracked this single fish by capturing the left and right images and then estimated the body length in full frames. Since the estimated body length of the fish is not consistent for each frame due to its continuous movement, we considered the final body length by getting the average of the body length for each frame.

The body length estimation of the single fish using 20 frames is in Figure 14, where the average body length is 20.895cm, with an average error of 2.38%. The maximum error is in Frame 2 at 5.52% error. The average body length is computed as \[ \frac{1}{n} \sum_{i=1}^{n} E_i \] while the average error is calculated using \[ \frac{1}{n} \sum_{i=1}^{n} E_i \] where $X_i$ is the estimated body length and $E_i$ is the error value for each frame. To calculate the length error value ($e$) for each frame $n$, we use

\[ e = \frac{\text{actual body length} - \text{estimated body length}}{\text{actual body length}} \times 100. \]

FIGURE 14. Tracking result of a single fish using 20 frames.

Figure 15 is an illustrative diagram of our proposed effective range filtering, executed after the 3D estimation. Figure 15(a) is the result of the instance segmentation from the collected video from Cage-1; (b) is the disparity image result, and the blue dots in (c) represent 12 segmented fish images from (a). The depth value of the fish is the 3D point cloud depth value where the fish is closest to the lens. Therefore, the effective range is filtered based on the depth value, and only the fish within the range of 50 - 200 cm will be estimated in terms of body length.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
The information provided in Table 3 shows the manual measurement of the body length of the fish and the estimation results, and its corresponding estimation error using 2D, 3D, and estimation using the most effective camera lens distance (50cm to 200cm), where the fish outside the effective range were discarded due to high error rate based on the earlier experiment that was carried out.

TABLE 3
MANUAL FISH SIZE MEASUREMENT AND THE DIFFERENT METHODS USED TO ESTIMATE THE FISH BODY LENGTH

| Fish Species | Videos | Measurement | 2D estimation | 3D estimation | Effective range 3D estimation |
|--------------|--------|-------------|---------------|---------------|------------------------------|
|              |        | body length (cm) | Est. value | Error | Est. value | Error | Est. value | Error |
| Porphyry seabream | A13-1 | 20.485 | 26.43 | 29.02 | 20.25 | 1.15 | 20.25 | 1.15 |
|              | A13-2 | 20.485 | 26.54 | 29.56 | 21.13 | 3.13 | 21.13 | 3.13 |
|              | A13-3 | 20.485 | 25.22 | 23.11 | 21.78 | 6.32 | 20.61 | 0.61 |
|              | A13-4 | 20.485 | 28.86 | 40.88 | 23.22 | 5.39 | 21.22 | 3.59 |
| Golden pomfret | Cage-1 | 30.3 | 43.62 | 43.3% | 37.82 | 24.61 | 28.37 | 4.71 |
|              | Cage-2 | 30.3 | 45.64 | 50.62% | 37.11 | 22.47% | 32.14 | 6.07% |
|              | Cage-3 | 32.374 | 47.41 | 46.48% | 37.22 | 14.97% | 31.34 | 5.19% |
|              | Cage-4 | 32.374 | 46.96 | 45.05% | 39.56 | 22.2% | 30.23 | 6.62% |

For the 2D measurement, we used the flat left and right images and combined them to generate a single image as the source image. The fish is first segmented, and the straight line of the segmented fish body part is used as the fish length. The fish length is calculated from its nose to the tail fork with the longest axis using the pixel-wise measurement of the segmented fish image. The 2D measurement has some problems since the body of the free-swimming fish is not straight, which makes the length measurement inaccurate. For the 3D measurement, we used the depth and 3D coordinate positions since there are instances where the fish body is in a curve form or different posture. We measured the length of the fish by the distance of the fish from the camera. The results show that the error of the estimated body length after the 3D reconstruction is significantly reduced compared to the results from the 2D estimation. Integrating the effective camera distance for the Porphyry seabream fish located in a smaller pond area seems not significant in the error reduction since it has almost the same results as the 3D estimation, unlike the results for the golden pomfret with a significant error reduction. The effectiveness of camera distance is more relevant to large fish cages, as manifested in the result for golden pomfret located in an open cage with more transparent water and a larger fish cage size. When the water is clear, the fish can still be seen even far from the stereo camera lens; thus, discarding the fish outside the effective camera distance will significantly reduce error for large cages.

V. CONCLUSIONS AND FUTURE WORKS
First, the deep neural network establishes the pixel correspondences between stereo images in real-time. Second, traditional deep learning models require large human-annotated labeled data for training. Our proposed approach is directly trained from the raw video data. With this, the training complexity of deep neural networks for 3D model construction is reduced. Third, establishing precise pixel correspondences from textureless stereo images using traditional stereo matching algorithms is very challenging. The interpolated signals of the matched object pairs ensure the correctness of the computed residual disparity image from precise correspondences with minimal object matching error. Lastly, the object-based stereo matching optimization algorithm is involved in the disparity image warping to minimize the resulting smoothness. The stereo camera can capture fish information from the underwater environment for fish body length estimation and fish weight conversion. This information can help establish the fish growth curve, and with the data on the feed amount, the meat exchange rate can be derived. Thus, farmers can optimize fish feeding, reduce production costs, and increase farm profit. The use of non-contact methods eases labor costs and prevents fish death and illnesses. This method also provides aquaculture farmers with the current condition of their fish products at any time. For the next step of our research, we plan to combine a stereo camera with a sonar system to generate a more accurate and precise method for measuring fish body length and weight. Fish species identification using the sonar enables the relative sonar to provide a depth reference value for 3D images, so combining them will improve the estimation accuracy.
REFERENCES

[1] W. Luo, A.G. Schwing and R. Urtasun, “Efficient Deep Learning for Stereo Matching," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 5695-5703.

[2] A. Milella, and G. Reina, “3D reconstruction and classification of natural environments by an autonomous vehicle using multi-baseline stereo”, Intelligent Service Robotics, vol. 7, pp. 79-92, 2014. DOI: 10.1007/s11370-014-0146-x.

[3] M. O’Byrne, V. Pakrashi, F. Schoefs, and B. Ghosh, “Semantic Segmentation of Underwater Imagery Using Deep Networks Trained on Synthetic Imagery”, Journal of Marine Science and Engineering, vol. 6, 93, 2018. DOI:10.3390/jmse6030093.

[4] M. Yang, J. Hu, C. Li, G. Rohde, Y. Du and K. Hu, “An In-Depth Survey of Underwater Image Enhancement and Restoration," IEEE Access, vol. 7, pp. 123638-123657, 2019.

[5] R. Schettini, and S. Corchs, “Underwater Image Processing: State of the Art of Restoration and Image Enhancement Methods”, EURASIP Journal on Advances in Signal Processing, 2010, DOI: 10.1155/2010/746052.

[6] B.G. Teo, and S. K. Dhillon, “An automated 3D modeling pipeline for constructing 3D models of MONOGENEAN HARDPART using machine learning techniques", BMC Bioinformatics, vol. 20, 2019. DOI: https://doi.org/10.1186/s12859-019-3210-x.

[7] W. Chen, J. Gao, H. Ling, E. Smith, J. Lehtinen, A. Jacobson, and S. Fidler, Learning to Predict 3D Objects with an Interpolation-based Differentiable Renderer, Presented at Annual Conference on Neural Information Processing System (NeurIps), 20109. arXiv:1908.01210.

[8] X. Han, H. Laga, and M. Bennamoun, “Image-Based 3D Object Reconstruction: State-of-the-Art and Trends in the Deep Learning Era”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 5, pp. 1578-1604, May 2021. DOI: 10.1109/TPAMI.2019.2954885.

[9] N. Ubiña, S.-C. Cheng, C.-C. Chang, and H.-Y. Chen, “Evaluating Fish Feeding Intensity in Aquaculture with Convolutional Neural Networks,” Aquaculture Engineering, vol. 94, August 2021. DOI: https://doi.org/10.1016/j.aquaeng.2021.102178.

[10] K. He, G. Gkioxari, P. Dollar, and R. Girshick, “Mask R-CNN,” 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 261-270. DOI: 10.1109/ICCV.2017.37.

[11] K.-J. Yoon and I. S. Kweon, “Adaptive support-weight approach for correspondence search,” in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 4, pp. 650-656, April 2006, DOI: 10.1109/TPAMI.2006.70.

[12] J. Sánchez Pérez, E. Meinhardt-Llopis, G. Facciolo, "TV-L1 Optical Flow Estimation" , Image Processing on Line, vol. 3, pp. 157-150, DOI: https://doi.org/10.5201/ipol.2013.26.

[13] H. Hirschmüller, "Accurate and efficient stereo processing by semi-global matching and mutual information," 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05), vol. 2, pp. 807-814. DOI: 10.1109/CVPR.2005.56.
Hsun-Yu Lan received the degree B.S. in Business Management from Cheng Shiu University, Kaohsiung, Taiwan, in 2007 and an M.S. degree in Aquaculture from National Taiwan Ocean University, Keelung, Taiwan, in 2021. She served as the Chairman’s Assistant of Long Dian Marine Biotechnology Co., Ltd. from 2014 to 2018. She is currently pursuing her Ph. D. degree in Aquaculture at the National Taiwan Ocean University. Her research interests include aquaculture economics, fish farm management, business management, and cross-disciplinary AI applications.

Hoang-Yang Lu received his B.S., M.S., and Ph.D. degrees from the National Taiwan University of Science and Technology, Taipei, in 1991, 1993, and 2007, respectively, all in electronics engineering. From 1993 to 2007, he was a lecturer at the Department of Electronics Engineering, Lee-Ming Institute of Technology, Taipei. He joined the Digital Technology Department, Kainan University, Taoyuan, Taiwan, in 2007 and the Department of Electronics Engineering, Huafan University, Taipei, in 2008, where he was an Associate Professor. Since 2009, he has been an Assistant Professor with the Department of Electrical Engineering, National Taiwan Ocean University (NTOU). At present, he is an Associate Professor at NTOU. His research interests include signal processing, communication systems, and soft computing.