

Article

An Underwater Multisensor Fusion Simultaneous Localization and Mapping System Based on Image Enhancement

Zeyang Liang¹, Kai Wang², Jiaqi Zhang¹ and Fubin Zhang^{3,*}

¹ Chinese Flight Test Establishment, Xi'an 710089, China

² Avic The First Aircraft Institute, Xi'an 710000, China

³ School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China

* Correspondence: zhangfubin@nwpu.edu.cn; Tel.: +86-1399-130-2665

Abstract: As a key method of ocean exploration, the positioning accuracy of autonomous underwater vehicles (AUVs) directly influences the success of subsequent missions. This study aims to develop a novel method to address the low accuracy in visual simultaneous localization and mapping (SLAM) within underwater environments, enhancing its application in the navigation and localization of AUVs. We propose an underwater multisensor fusion SLAM system based on image enhancement. First, we integrate hybrid attention mechanisms with generative adversarial networks to address the blurring and low contrast in underwater images, thereby increasing the number of feature points. Next, we develop an underwater feature-matching algorithm based on a local matcher to solve the feature tracking problem caused by grayscale changes in the enhanced image. Finally, we tightly couple the Doppler velocity log (DVL) with the SLAM algorithm to better adapt to underwater environments. The experiments demonstrate that, compared to other algorithms, our proposed method achieves reductions in both mean absolute error (MAE) and standard deviation (STD) by up to 68.18% and 44.44%, respectively, when all algorithms are operating normally. Additionally, the MAE and STD of our algorithm are 0.84 m and 0.48 m, respectively, when other algorithms fail to operate properly.

Keywords: underwater visual SLAM; multisensor fusion; image enhancement; hybrid attention



Citation: Liang, Z.; Wang, K.; Zhang, J.; Zhang, F. An Underwater Multisensor Fusion Simultaneous Localization and Mapping System Based on Image Enhancement. *J. Mar. Sci. Eng.* **2024**, *12*, 1170. <https://doi.org/10.3390/jmse12071170>

Academic Editor: Alessandro Ridolfi

Received: 16 May 2024

Revised: 2 July 2024

Accepted: 9 July 2024

Published: 12 July 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Since the twenty-first century, due to the increasing shortage of land resources, the ocean, as a newly developed treasure trove for humanity, has emerged as a new world center [1,2]. Due to the complexity of the marine environment, the cost of manual operation in this environment is high, and there are various hidden dangers. Autonomous underwater vehicles (AUVs) [3,4], owing to their autonomous navigation capabilities, can effectively replace manual underwater operations such as seabed exploration, seabed archaeology, and seabed minesweeping [5,6], and have gradually become essential tools for executing underwater tasks. The prerequisite for AUVs to perform underwater missions is the availability of a high-precision navigation and localization system. Hydroacoustic localization systems [7,8], inertial navigation and localization systems [9,10], and simultaneous localization and mapping (SLAM) technology [11,12] have been proposed as solutions to the underwater navigation challenges faced by AUVs.

The process of the hydroacoustic positioning system is summarized as the interaction between an acoustic transmitting transducer and a receiving transducer to determine the target position. As hydroacoustic positioning systems generally use acoustic waves as the communication carrier, they are affected by the underwater environment, leading to increased noise in the hydroacoustic signal, longer communication delays, and other issues that significantly impact the accuracy and real-time capabilities of navigation and positioning. Additionally, hydroacoustic positioning systems are expensive and difficult to install, further limiting their application scenarios.

Inertial navigation and positioning systems achieve a high degree of covertness and autonomy by using the acceleration and angular velocity data obtained from accelerometers and gyroscopes to perform dead reckoning, thereby determining the current position. In practical applications, this information is generally provided by inertial measurement devices. High-precision navigation equipment is expensive and generally not suitable for civilian use, while low-cost micro-electro-mechanical system inertial measurement units (MEMS IMUs) are less accurate and cannot be used independently for extended periods.

With the development of computer technology, SLAM technology with a laser or camera as the core sensor is increasingly employed. Laser SLAM [13,14], which achieves navigation and positioning by comparing the emitted and reflected laser beams, is gradually being applied in underwater environments. For example, the literature [15,16] proposes the use of single-photon underwater LiDAR for bathymetric measurements, and the measurement results are highly consistent with synchronized sonar data. Visual SLAM [17,18], which determines position through image data, has the advantages of no drift and high stability, and can provide high-precision navigation and positioning for AUVs.

Currently, SLAM algorithms applied to AUVs are broadly classified into two categories: EKF-SLAM, based on the extended Kalman filter (EKF); fast SLAM, based on the particle filter (PF); and graph-SLAM, based on graph optimization.

The computation of EKF-SLAM [19] is divided into four processes: prediction, observation update, data association, and state dimension expansion. A study [20] proposed an algorithm for underwater localization using acoustic signals based on the EKF-SLAM algorithm, which fuses inertial sensors and the direction angle of the acoustic source obtained from Bayesian estimation for position estimation. Authors [21] developed an enhanced EKF-SLAM, and the algorithm uses position error models based on side-scan sonar and AUV inertial navigation systems as input data, rather than directly using the data from inertial sensors and side-scan sonar. EKF-SLAM inherits the advantages and disadvantages of the EKF algorithm. The algorithm's principle is simple and easy to implement, but the error caused by system nonlinearity may lead to nonconvergence and high computational cost.

The basis of FAST-SLAM is the PF, whose core idea is to use a series of randomly drawn samples and their weights to represent the posterior probability distribution of the state. When the number of samples is sufficiently large, the true posterior distribution can be well approximated by such random sampling. Researchers [22] utilized the mean trajectory graph, a method that reduces computational consumption by retaining only the current estimated position of the AUV in its particles, while all historical states of the AUV are stored in the mean trajectory graph. Others [23] proposed a particle-filter-based bathymetric simultaneous localization and mapping (BSLAM) method with mean trajectory map representation, where the particles of the algorithm retain only the current estimated position of the AUV, while all historical states of the AUV are stored in the averaged trajectory map. The disadvantages of FAST-SLAM include the problem of particle "degeneracy" and the difficulty of finding a general selection criterion for the importance density function used to generate particles.

In graph-SLAM, the position of an AUV is represented as a node or vertex, and the relationships between positions constitute edges. The process is mainly divided into two parts: the front end, responsible for processing the data collected by the sensors, and the back end, which uses a nonlinear optimization algorithm to determine the robot's position. The literature [24] proposes a collaborative SLAM framework where each AUV generates its own local map while using marginalized pose and sparsified information matrices to reduce the size of communication packets, ultimately achieving collaborative localization and mapping. In a study [25], to enhance the robustness of SLAM, a robust estimator that fuses the robot's kinematic model with proprioceptive sensors was proposed to propagate the pose when the visual inertial odometer (VIO) fails, ensuring proper localization. Graph-optimization-based algorithms were previously considered too time consuming to meet the real-time requirements of SLAM. However, with the emergence of efficient solving

algorithms and the rapid development of related hardware, graph-optimization-based algorithms have regained attention and become a hot topic in the current SLAM research.

The disadvantages of underwater images [26,27], such as color deviation and blurring, pose the risk of crashing the SLAM operation. In this study, to meet the demand for AUV underwater navigation and localization, an underwater multisensor fusion SLAM algorithm based on image enhancement was designed by integrating an underwater image enhancement module into VINS-Mono [28], which belongs to the graph-SLAM category, and fusing data from various sensors such as a monocular camera, IMU, and Doppler velocity log (DVL).

The main contributions of this study are summarized as follows:

- In this paper, an underwater image enhancement algorithm based on a generative adversarial network [29] is proposed. To improve the quality of underwater images, we designed a hybrid attention module, consisting of channel attention and spatial attention, and applied it to the generator to enhance the underwater image enhancement effect. Additionally, we constructed a multinomial loss function to improve training efficiency.
- In this paper, a multisensor fusion SLAM algorithm is proposed, based on the VINS-Mono framework, incorporating DVL, and making corresponding improvements to its measurement preprocessing, initialization, and nonlinear optimization components. Additionally, to address the impact of grayscale changes caused by image enhancement on image matching, this paper proposes an underwater image matching algorithm based on a local matcher.

In Section 2, we describe the specific algorithms used in this study. The results obtained from the system are presented in Section 3. Finally, we provide a brief discussion in Section 4 and the conclusions in Section 5.

2. Materials and Methods

In this section, we describe the programs and methods used in this study, divided into the following two aspects: First, the underwater image enhancement algorithm based on a generative adversarial network is presented. Second, the integration of DVL into the VINS-Mono algorithm and the underwater feature-matching algorithm based on the local matcher are introduced.

The algorithm proposed in this paper is based on the traditional VINS-Mono framework and consists of four parts: measurement preprocessing, initialization, nonlinear optimization, and loop closure. The system framework of the proposed algorithm is shown in Figure 1.

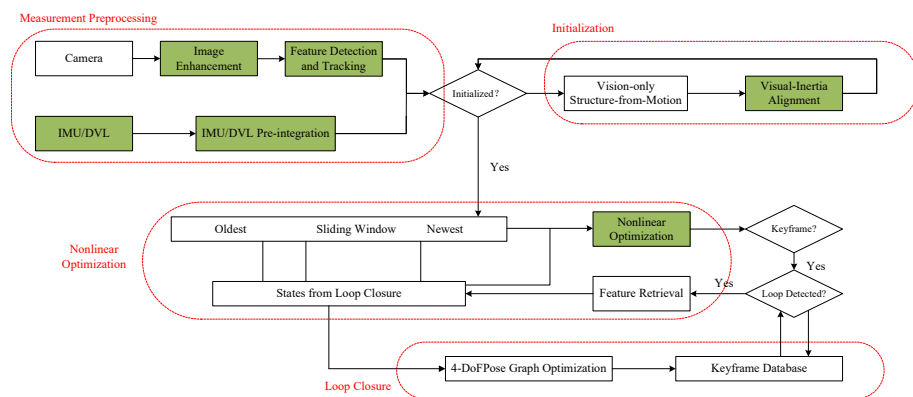


Figure 1. The system framework for the proposed algorithm. The color-coded parts indicate the components that were improved in this study.

2.1. Underwater Image Enhancement Algorithm Based on Generative Adversarial Network

The disadvantages of underwater images, such as partial color, blurring, and low contrast, lead to a significant reduction in the number of extractable feature points, which in turn can cause the collapse of the entire SLAM system. To address this problem, this subsection proposes a generative adversarial network based on a hybrid attention mechanism to enhance underwater images. The hybrid attention module, which serves as the core part of the generator, aims to restore the color and texture of the image and consists of a channel attention module and a spatial attention module. The overall network framework is a generative adversarial network, with the adversarial network framework based on U-Net [30]. The generator network framework and hybrid attention module network framework are shown in Figure 2.

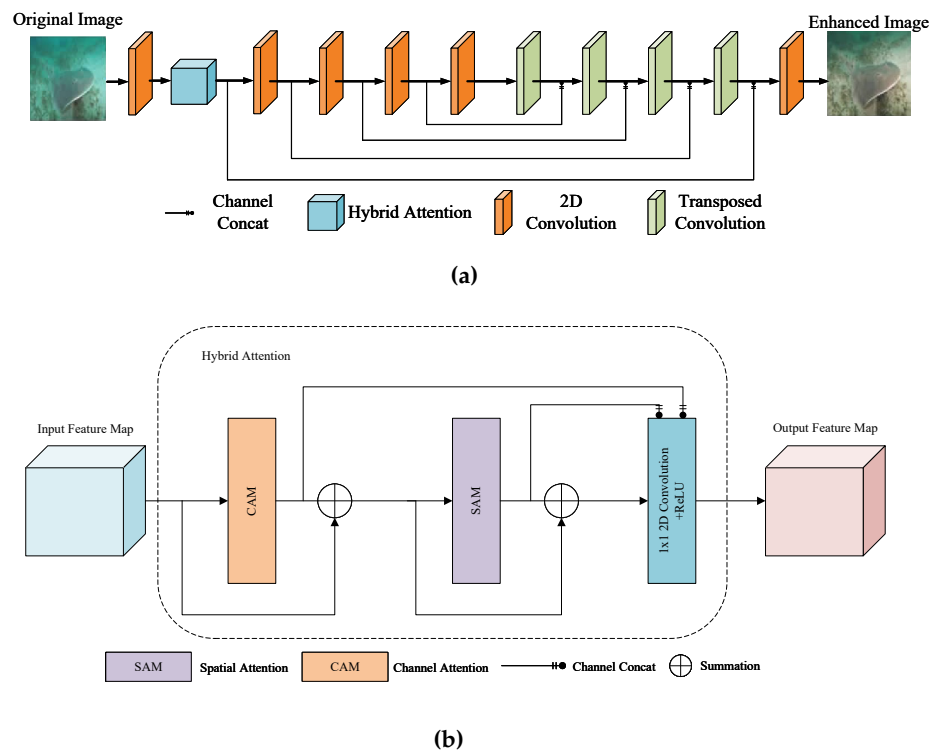


Figure 2. Network framework: (a) generator network framework; (b) hybrid attention module network framework.

2.1.1. Channel Attention Module

The channel attention module reallocates channel resources to aid in the color restoration of images. It aggregates feature map information using both maximum pooling and average pooling and achieves adaptive selection through learned parameters, ensuring that the two pooling results have different weights to obtain better channel attention maps. The network framework is shown in Figure 3, and the specific process is detailed below:

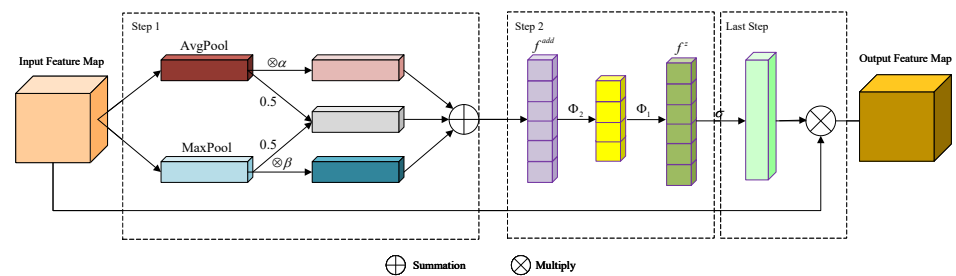


Figure 3. Channel attention module network framework.

Step 1: The input feature map $X \in \mathbb{R}^{C_{in} \times H_{in} \times W_{in}}$ undergoes two pooling operations in two pooling layers. The pooling results are then weighted by the learning parameters α and β and summed element-wise with their mean to obtain the feature tensor $f^{add} \in \mathbb{R}^{C_{in} \times 1 \times 1}$. This process can be expressed by the following equation:

$$f^{add} = (\alpha \otimes AvgPool(X)) \oplus (\beta \otimes MaxPool(X)) \oplus (0.5 \otimes (AvgPool(X) \oplus MaxPool(X))) \quad (1)$$

where *AvgPool* represents average pooling; *MaxPool* represents max pooling; \oplus denotes element-wise summation; and \otimes denotes element-wise multiplication. The parameters α and β belong to $(0, 1]$, and the network achieves adaptive tuning of channel weights through these two learning parameters.

Step 2: The feature tensor f^{add} undergoes dimensionality reduction followed by dimensionality enhancement to achieve cross-channel information interaction, resulting in the feature tensor $f^z \in \mathbb{R}^{C_{in} \times 1 \times 1}$. The process can be expressed by the following equation:

$$f^z = \Phi_1(\Phi_2(f^{add})) = C2D_{1 \times 1}^{C_{in} \times d}(\delta(BN(C2D_{1 \times 1}^{d \times C_{in}}(f^{add})))) \quad (2)$$

where Φ_1 represents the ascending operation; Φ_2 represents the descending operation; $C2D_{1 \times 1}^{C_{in} \times d}$ denotes a 1×1 2D convolution with the number of output channels as C_{in} and the number of input channels as d ; $C2D_{1 \times 1}^{d \times C_{in}}$ denotes a 1×1 2D convolution with the number of output channels as d and the number of input channels as C_{in} ; *BN* represents batch normalization; and δ is the *ReLU* activation function. The parameter d is determined by the hyperparameter r :

$$d = \max(C_{in} / r, L) \quad (3)$$

In the final step, the feature tensor f^z is normalized and multiplied element-wise with the input feature map X to obtain the output feature map $Y \in \mathbb{R}^{C_{in} \times H_{in} \times W_{in}}$:

$$Y = \sigma(f^z) \otimes X \quad (4)$$

where σ is the *softmax* activation function.

2.1.2. Spatial Attention Module

The spatial attention module strengthens the network’s learning of information-rich regions, aiding in the recovery of the highly textured parts of the image, and determines the importance of the features in each part of the different channels through learning. To ensure the network pays more attention to the highly textured parts of the image, this paper proposes a feature separation algorithm that divides the features in each channel into important and minor features. The spatial attention module first uses the proposed algorithm to achieve feature separation, then applies average pooling and max pooling operations along the channel dimensions, and finally passes the two pooling results through a series of convolutional layers to obtain the final spatial attention map. The network framework is shown in Figure 4, and the specific flow is detailed below:

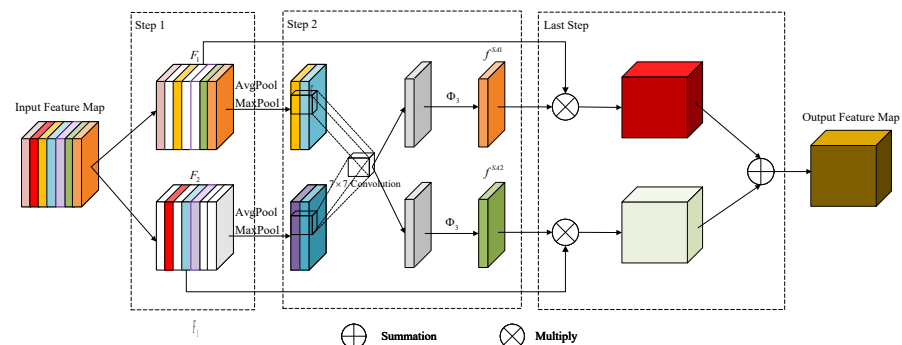


Figure 4. Spatial attention module network framework.

Step 1: The input feature map $X \in \mathbb{R}^{C_{in} \times H_{in} \times W_{in}}$ undergoes adaptive average pooling to obtain the average value of each layer. The part above the average value is defined as important features, and the part below the average value is defined as secondary features. Two tensors are then defined to store the results: one tensor sets the important features to 1 and the secondary features to 0, while the other tensor does the opposite. The two tensors are then multiplied element-wise with the input feature map to obtain the important feature map $F_1 \in \mathbb{R}^{C_{in} \times H_{in} \times W_{in}}$ and the secondary feature map $F_2 \in \mathbb{R}^{C_{in} \times H_{in} \times W_{in}}$. The Python code is shown in Algorithm 1.

Algorithm 1 Python code for feature separation.

Input: X
Output: F_1, F_2

```

1: def channel_separation(X):
2:     one = torch.ones_like(X)
3:     zero = torch.zeros_like(X)
4:     avg_pool = nn.AdaptiveAvgPool2d(1)
5:     avg = avg_pool(X)
6:     important_tensor = torch.where(X > avg, one, zero)
7:     subimportant_tensor = 1 - important_tensor
8:     F1 = X * important_tensor
9:     F2 = X * subimportant_tensor
10:    return F1, F2

```

Step 2: The feature maps F_1 and F_2 undergo a pooling layer, a 7×7 shared convolutional layer and a series of nonlinear operations to obtain the corresponding spatial attention tensors $f^{SA1} \in \mathbb{R}^{1 \times H_{in} \times W_{in}}$ and $f^{SA2} \in \mathbb{R}^{1 \times H_{in} \times W_{in}}$. The process can be expressed as follows:

$$\begin{aligned} f^{SA1} &= \Phi_3 \left(C2D_{7 \times 7}^{1 \times 1}([AvgPool(F_1); MaxPool(F_1)]) \right) \\ f^{SA2} &= \Phi_3 \left(C2D_{7 \times 7}^{1 \times 1}([AvgPool(F_2); MaxPool(F_2)]) \right) \end{aligned} \quad (5)$$

where Φ_3 represents a series of nonlinear operations consisting of a sequence of BN layers and ReLU functions; and $C2D_{7 \times 7}^{1 \times 1}$ denotes a 7×7 2D convolution with 1 output channels and 1 input channel.

In the final step, the spatial attention tensors f^{SA1} and f^{SA2} are multiplied with the feature maps F_1 and F_2 , respectively, and the results are summed element-wise to obtain the output feature map $Y \in \mathbb{R}^{C_{in} \times H_{in} \times W_{in}}$. The process can be expressed as follows:

$$Y = (f^{SA1} \otimes F_1) \oplus (f^{SA2} \otimes F_2) \quad (6)$$

2.1.3. Loss Function

In this method, the loss function L is obtained by linearly weighting the adversarial loss L_{adv} [29], the L1 loss L_1 [31], the gradient loss L_{gra} , and the color bias loss L_{col} . The specific form of the loss function is shown in the following equation:

$$L = L_{adv} + \lambda_{g1}L_1 + \lambda_{g2}L_{gra} + \lambda_{g3}L_{col} \quad (7)$$

where the scaling factors λ_{g1} , λ_{g2} , and λ_{g3} are set to 0.3, 0.3, and 0.4

Among them, the gradient loss L_{gra} can represent the clarity and integrity at the image boundary to a certain extent, aiding in the generation of image details, as follows:

$$L_{gra} = E_{x,y} \{ \|\nabla_i y - \nabla_i [G(x)]\|_1 \} + E_{x,y} \{ \|\nabla_j y - \nabla_j [G(x)]\|_1 \} \quad (8)$$

where ∇_i, ∇_j are the horizontal and vertical gradient operators.

The color bias loss L_{col} is used to encourage the color distribution of the generated image to align more closely with the real image by measuring the degree of color bias in the generated image, as follows:

$$\begin{aligned}
 L_{col} &= D(y)/M(y) - D(x)/M(x) \\
 d_a &= \frac{\sum_{i=1}^M \sum_{i=1}^N a}{MN}, d_b = \frac{\sum_{i=1}^M \sum_{i=1}^N b}{MN} \\
 D &= \sqrt{d_a^2 + d_b^2} \\
 M_a &= \frac{\sum_{i=1}^M \sum_{i=1}^N (a-d_a)}{MN}, M_b = \frac{\sum_{i=1}^M \sum_{i=1}^N (d-d_b)}{MN} \\
 M &= \sqrt{M_a^2 + M_b^2}
 \end{aligned} \tag{9}$$

where D denotes the average color difference of the image; M denotes the center distance of the image; a and b denote the two-dimensional coordinates of the a channel and b channel of the image in Lab space; and M and N denote the maximum values of the two-dimensional coordinates of the image in Lab space.

2.2. Underwater Multisensor Fusion SLAM Algorithm

2.2.1. DVL Tightly Coupled to SLAM Algorithm

In this paper, $(\cdot)^w$ represents the world coordinate system, $(\cdot)^b$ represents the body coordinate system (defined the same as the IMU coordinate system), $(\cdot)^c$ represents the camera coordinate system, and $(\cdot)^d$ represents the DVL coordinate system. Meanwhile, the rotation matrix R and quaternion q are used to represent rotation.

Adding DVL to VINS-Mono requires corresponding adjustments to pr-integration, initialization, and nonlinear optimization. The following sections describe each of these adjustments in detail.

Based on the IMU kinematic model [32], by adding the DVL information, the kinematic IMU/DVL model in the world coordinate system can be obtained as shown in Equation (10):

$$\begin{aligned}
 \mathbf{p}_{b_{k+1}}^w &= \mathbf{p}_{b_k}^w + \mathbf{v}_{b_k}^w \Delta t + \int \int_{t \in [k, k+1]} [\mathbf{R}_t^w (\hat{\mathbf{a}}_t - \mathbf{b}_a) - \mathbf{g}^w] dt^2 \\
 \mathbf{v}_{b_{k+1}}^w &= \mathbf{v}_{b_k}^w + \int_{t \in [k, k+1]} [\mathbf{R}_t^w (\hat{\mathbf{a}}_t - \mathbf{b}_a) - \mathbf{g}^w] dt \\
 \mathbf{q}_{b_{k+1}}^w &= \mathbf{q}_{b_k}^w \odot \int_{t \in [k, k+1]} \frac{1}{2} \Omega(\hat{\boldsymbol{\omega}}_t - \mathbf{b}_\omega) \mathbf{q}_t^{b_k} dt \\
 \mathbf{p}_{d_{k+1}}^w &= \mathbf{p}_{d_k}^w + \int_{t \in [k, k+1]} [\mathbf{R}_t^w \mathbf{R}_d^b (\hat{\mathbf{v}}_t - \mathbf{b}_d)] dt
 \end{aligned} \tag{10}$$

where $\mathbf{p}_b^w, \mathbf{v}_b^w, \mathbf{q}_b^w$ denote the position, velocity, and rotation of the IMU in the world coordinate system; \mathbf{p}_d^w is the position of the DVL in the world coordinate system; $\hat{\mathbf{a}}_t, \hat{\boldsymbol{\omega}}_t, \hat{\mathbf{v}}_t$ are the measured values of the accelerometer, gyroscope, and DVL, respectively; $\mathbf{b}_a, \mathbf{b}_\omega, \mathbf{b}_d$ are the corresponding zero biases; Δt denotes the time difference between the two frames; \mathbf{R}_t^w is the rotation of the IMU in the world coordinate system at time t ; \mathbf{g}^w is the acceleration of gravity in the world coordinate system; $\mathbf{q}_t^{b_k}$ is the rotation increase in the IMU in the b_k frame at time t ; \mathbf{R}_d^b is the rotational parameter of IMU-DVL; \odot is the symbol for quadratic multiplication; and Ω is defined as:

$$\Omega(\boldsymbol{\omega}) = \begin{bmatrix} -[\boldsymbol{\omega}]_\times & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^T & 0 \end{bmatrix}, [\boldsymbol{\omega}]_\times = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & -\omega_x & 0 \end{bmatrix} \tag{11}$$

Equation (10) is multiplied by $R_w^{b_k}$ on both sides to obtain the IMU pre-integration $\alpha_{b_{k+1}}^{b_k}, \beta_{b_{k+1}}^{b_k}, \gamma_{b_{k+1}}^{b_k}$ and the DVL pre-integration $\eta_{d_{k+1}}^{b_k}$:

$$\begin{aligned} \alpha_{b_{k+1}}^{b_k} &= \iint_{t \in [k, k+1]} \left[R_t^{b_k} (\hat{a}_t - b_a) \right] dt^2 \\ \beta_{b_{k+1}}^{b_k} &= \int_{t \in [k, k+1]} \left[R_t^{b_k} (\hat{a}_t - b_a) \right] dt \\ \gamma_{b_{k+1}}^{b_k} &= \int_{t \in [k, k+1]} \frac{1}{2} \Omega (\hat{\omega}_t - b_\omega) \gamma_t^{b_k} dt \\ \eta_{d_{k+1}}^{b_k} &= \int_{t \in [k, k+1]} \left[R_t^{b_k} R_d^b (\hat{v}_t - b_d) \right] dt \end{aligned} \tag{12}$$

where $R_w^{b_k} = (R_{b_k}^w)^T$, $R_{b_k}^w$ represents the rotation of the IMU in the world coordinate system for the k frame.

The first-order Taylor series expansion of Equation (12) is given by

$$\begin{aligned} \alpha_{b_{k+1}}^{b_k} &\approx \hat{\alpha}_{b_{k+1}}^{b_k} + J_{b_a}^\alpha \delta b_a + J_{b_\omega}^\alpha \delta b_\omega \\ \beta_{b_{k+1}}^{b_k} &\approx \hat{\beta}_{b_{k+1}}^{b_k} + J_{b_a}^\beta \delta b_a + J_{b_\omega}^\beta \delta b_\omega \\ \gamma_{b_{k+1}}^{b_k} &\approx \hat{\gamma}_{b_{k+1}}^{b_k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2} J_{b_\omega}^\gamma \delta b_\omega \end{bmatrix} \\ \eta_{d_{k+1}}^{b_k} &\approx \hat{\eta}_{d_{k+1}}^{b_k} + J_{b_d}^\eta \delta b_d + J_{b_\omega}^\eta \delta b_\omega \end{aligned} \tag{13}$$

where $\hat{\alpha}_{b_{k+1}}^{b_k}, \hat{\beta}_{b_{k+1}}^{b_k}, \hat{\gamma}_{b_{k+1}}^{b_k}, \hat{\eta}_{d_{k+1}}^{b_k}$ are the IMU/DVL preintegrated measurements; $\delta b_a, \delta b_\omega, \delta b_d$ are the zero-bias errors of the accelerometer, gyroscope, and DVL; and J denotes the respective Jacobian matrix.

The estimation of the rotational parameter R_c^b and the gyroscope zero bias during initialization is consistent with that of VINS-Mono. Therefore, only the estimation of the accelerometer's zero bias and the DVL's zero bias is presented in this paper. Considering scale s , the pose of the IMU/DVL in the world coordinate system is

$$\begin{aligned} R_b^w &= R_c^w (R_c^b)^{-1} \\ s p_b^w &= s p_c^w - R_b^w p_c^b \\ s p_d^w &= s p_c^w - R_b^w R_d^b p_c^d \end{aligned} \tag{14}$$

where R_c^w and p_c^w are the attitude and position of the camera in the world coordinate system; p_c^b and p_c^d are the translational parameters of the IMU-camera and DVL-camera; and R_d^b is the rotational parameter of the IMU-DVL.

The state variables are selected as $\chi_0 = [v_{b_0}^{b_0}, \dots, v_{b_k}^{b_k}, v_{b_{k+1}}^{b_{k+1}}, g^w, s]$; Equation (10) and Equation (14) are combined to obtain Equation (15), which is solved to obtain the external parameter estimates of velocity, gravity vector, and scale factor $\hat{\chi}_0 = [\hat{v}_{b_0}^{b_0}, \dots, \hat{v}_{b_k}^{b_k}, \hat{v}_{b_{k+1}}^{b_{k+1}}, \hat{g}^w, \hat{s}]$.

$$\begin{aligned} &\begin{bmatrix} -I \Delta t & 0 & \frac{1}{2} R_w^{b_k} \Delta t^2 & 2 R_w^{b_k} (p_{b_{k+1}}^w - p_{b_k}^w) \\ -I & R_w^{b_k} R_{b_{k+1}}^w & R_w^{b_k} \Delta t & 0 \end{bmatrix} \begin{bmatrix} v_{b_k}^{b_k} \\ v_{b_{k+1}}^{b_{k+1}} \\ g^w \\ s \end{bmatrix} \\ &= \begin{bmatrix} \hat{\alpha}_{b_{k+1}}^{b_k} + \hat{\eta}_{b_{k+1}}^{b_k} - (I - R_w^{b_k} R_{b_{k+1}}^w) (p_c^b + R_d^b p_c^d) \\ \hat{\beta}_{b_{k+1}}^{b_k} \end{bmatrix} \end{aligned} \tag{15}$$

To reduce the effect of gravity on the estimated accelerometer zero bias and DVL zero bias, a gravity refinement step is required.

$$g^w = g \frac{\hat{g}^w}{\|\hat{g}^w\|} + [b_1 \quad b_2] \begin{bmatrix} m_1 \\ m_2 \end{bmatrix} = g \frac{\hat{g}^w}{\|\hat{g}^w\|} + b m \tag{16}$$

where \mathbf{b}_1 and \mathbf{b}_2 are the two orthogonal bases in the gravity tangent plane, and m_1 and m_2 are their respective lengths.

The state variables are selected as $\chi_0 = [\mathbf{v}_{b_0}^{b_0}, \dots, \mathbf{v}_{b_k}^{b_k}, \mathbf{v}_{b_{k+1}}^{b_{k+1}}, \mathbf{m}, s, \delta \mathbf{b}_a, \delta \mathbf{b}_d]$. Equations (13), (15), and (16) are combined to obtain Equation (17), which is solved to obtain the external parameter estimates of the zero bias of the accelerometer and the zero bias of the DVL.

$$\begin{aligned} & \begin{bmatrix} -I\Delta t & 0 & \frac{1}{2}\mathbf{R}_w^{b_k}\mathbf{b}\Delta t^2 & 2\mathbf{R}_w^{b_k}(\mathbf{p}_{c_{k+1}}^w - \mathbf{p}_{c_k}^w) & -\mathbf{J}_{b_a}^\alpha & -\mathbf{J}_{b_d}^\eta \\ -I & \mathbf{R}_w^{b_k}\mathbf{R}_{b_{k+1}}^w & \mathbf{R}_w^{b_k}\mathbf{b}\Delta t & 0 & -\mathbf{J}_{b_a}^\beta & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v}_{b_k}^{b_k} \\ \mathbf{v}_{b_{k+1}}^{b_{k+1}} \\ \mathbf{m} \\ s \\ \delta \mathbf{b}_a \\ \delta \mathbf{b}_d \end{bmatrix} \\ & = \begin{bmatrix} \hat{\mathbf{a}}_{b_{k+1}}^{b_k} + \hat{\boldsymbol{\eta}}_{b_{k+1}}^{b_k} - \left(\mathbf{I} - \mathbf{R}_w^{b_k}\mathbf{R}_{b_{k+1}}^w\right)\left(\mathbf{p}_c^b + \mathbf{R}_d^b\mathbf{p}_d^b\right) - \frac{1}{2}\mathbf{R}_w^{b_k}\frac{\hat{\mathbf{g}}^w}{\|\hat{\mathbf{g}}^w\|}\mathcal{G}\Delta t^2 \\ \hat{\boldsymbol{\beta}}_{b_{k+1}}^{b_k} - \mathbf{R}_w^{b_k}\frac{\hat{\mathbf{g}}^w}{\|\hat{\mathbf{g}}^w\|}\mathcal{G}\Delta t \end{bmatrix} \end{aligned} \quad (17)$$

After obtaining the accelerometer's zero bias and DVL's zero bias, they are repropagated for all IMU/DVL preintegrations, thereby completing the initialization of the SLAM algorithm presented in this paper.

For nonlinear optimization, the cost function after adding DVL is

$$\min_{\chi} \left\{ \sum_{(l,j) \in C} \left\| \mathbf{r}_C(\hat{\mathbf{z}}_l^{c_j}, \chi) \right\|_{\mathbf{P}_l^{c_j}}^2 + \sum_{k \in BO} \left\| \mathbf{r}_{BO}(\hat{\mathbf{z}}_{b_{k+1}}^{b_k}, \chi) \right\|_{\mathbf{P}_{b_{k+1}}^{b_k}}^2 + \left\| \mathbf{r}_p - \mathbf{J}_p\chi \right\|^2 \right\} \quad (18)$$

where $\mathbf{r}_C(\hat{\mathbf{z}}_l^{c_j}, \chi)$, $\mathbf{r}_{BO}(\hat{\mathbf{z}}_{b_{k+1}}^{b_k}, \chi)$, $\{\mathbf{r}_p, \mathbf{J}_p\}$ are the visual residuals, IMU/DVL preintegrated residuals, and marginalization constraint. The visual residuals and marginalized residuals are consistent with VINS-Mono, and the IMU/DVL preintegrated residuals is

$$\mathbf{r}_{BO}(\hat{\mathbf{z}}_{b_{k+1}}^{b_k}, \chi) = \begin{bmatrix} \delta \boldsymbol{\alpha}_{b_{k+1}}^{b_k} \\ \delta \boldsymbol{\beta}_{b_{k+1}}^{b_k} \\ \delta \boldsymbol{\gamma}_{b_{k+1}}^{b_k} \\ \delta \boldsymbol{\eta}_{d_{k+1}}^{b_k} \\ \delta \mathbf{b}_a \\ \delta \mathbf{b}_\omega \\ \delta \mathbf{b}_d \end{bmatrix} = \begin{bmatrix} \mathbf{R}_w^{b_k}(\mathbf{p}_{b_{k+1}}^w - \mathbf{p}_{b_k}^w + \frac{1}{2}\mathbf{g}^w\Delta t^2 - \mathbf{v}_{b_k}^w\Delta t) - \hat{\boldsymbol{\alpha}}_{b_{k+1}}^{b_k} \\ \mathbf{R}_w^{b_k}(\mathbf{v}_{b_{k+1}}^w + \mathbf{g}^w\Delta t - \mathbf{v}_{b_k}^w) - \hat{\boldsymbol{\beta}}_{b_{k+1}}^{b_k} \\ 2 \left[\mathbf{q}_{b_k}^{w-1} \otimes \mathbf{q}_{b_{k+1}}^w \otimes \hat{\boldsymbol{\gamma}}_{b_{k+1}}^{b_k} \right]_{xyz}^{-1} \\ \mathbf{R}_w^{b_k}(\mathbf{p}_{b_{k+1}}^w - \mathbf{p}_{b_k}^w) + (\mathbf{R}_w^{b_k}\mathbf{R}_{b_{k+1}}^w - \mathbf{I})\mathbf{p}_d^b - \hat{\boldsymbol{\eta}}_{d_{k+1}}^{b_k} \\ \mathbf{b}_{a_{k+1}} - \mathbf{b}_{a_k} \\ \mathbf{b}_{\omega_{k+1}} - \mathbf{b}_{\omega_k} \\ \mathbf{b}_{d_{k+1}} - \mathbf{b}_{d_k} \end{bmatrix} \quad (19)$$

where $\delta \boldsymbol{\alpha}_{b_{k+1}}^{b_k}$, $\delta \boldsymbol{\beta}_{b_{k+1}}^{b_k}$, $\delta \boldsymbol{\gamma}_{b_{k+1}}^{b_k}$, $\delta \boldsymbol{\eta}_{d_{k+1}}^{b_k}$ are the measurements residual of IMU/DVL pre-integration; $\delta \mathbf{b}_a$, $\delta \mathbf{b}_\omega$, $\delta \mathbf{b}_d$ are the zero bias residual of IMU/DVL.

2.2.2. Underwater Feature-Matching Algorithm Based on a Local Matcher

After the underwater images undergo image enhancement, although their clarity and contrast improve, the grayscale also changes to some extent, as shown in Figure 5, thus putting the optical flow algorithm [33] at risk of failure. To solve this problem, this paper proposes an underwater feature-matching algorithm based on a local matcher.

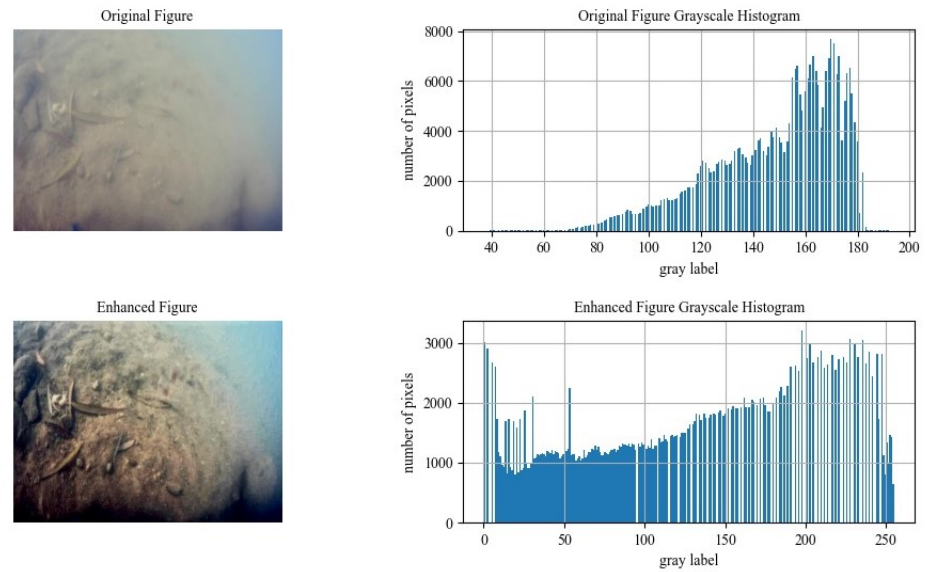


Figure 5. Grayscale comparison before and after underwater image enhancement.

Assuming that the two consecutive frames of the image are the c_k frame and the c_{k+1} frame, the corresponding IMU measurements are the b_k frame and the b_{k+1} frame, and the DVL measurements are the v_k frame and the v_{k+1} frame. The relative position Δp and relative attitude Δq between the two frames can be obtained according to the following equation:

$$\begin{aligned} \Delta p &= R_d^c \left[\frac{(v_k - b_d) + (v_{k+1} - b_d)}{2} \Delta t \right] \\ \Delta q &= \int_{t \in [k, k+1]} \frac{1}{2} R_b^c \Omega(\hat{\omega}_t - b_\omega) q_t^{b_k} dt \end{aligned} \quad (20)$$

where R_b^c and R_d^c are the rotational parameters for the camera-IMU and camera-DVL.

Assume that the normalized coordinates corresponding to the projections of a way-point on the images of the c_k frame and the c_{k+1} frame are p^{c_k} and $p^{c_{k+1}}$, the camera's internal reference is K and the pixel coordinates corresponding to p^{c_k} is u^{c_k} . The predicted value $\hat{u}^{c_{k+1}}$ of the pixel coordinate $p^{c_{k+1}}$ corresponding to $u^{c_{k+1}}$ can be obtained using the following equation:

$$\begin{aligned} p^{c_k} &= K^{-1} u^{c_k} \\ p^{c_{k+1}} &= \Delta R p^{c_k} + \Delta p \\ \hat{u}^{c_{k+1}} &= K p^{c_{k+1}} \end{aligned} \quad (21)$$

where ΔR is the matrix form of q .

To apply the predicted value $\hat{u}^{c_{k+1}}$ to feature matching, a local matcher with a search window was designed. First, a search window is created, which is a circular or square area centered on the predicted value $\hat{u}^{c_{k+1}}$, and its radius varies from small to large. Next, candidate features are searched within the small radius area. If the number of candidates exceeds the preset threshold (20 features), then the 20 features with the maximum response value are retained. Otherwise, the radius of the search window is enlarged, and the above search steps are repeated until 20 candidate points are detected or the detection area reaches or exceeds the size of the entire image. Finally, the feature points in the search window are regarded as potential matching candidates, and the feature point with the closest descriptor distance is obtained by brute-force matching in a small range.

Through the local matcher, even if there is a certain discrepancy in the enhancement effect between the two consecutive frames, it can still ensure that the feature points of the former image have a candidate match in the latter image. Then, after applying the RANSAC algorithm [34], a better matching result can be obtained. The flowchart of the underwater feature-matching algorithm is shown in Figure 6.

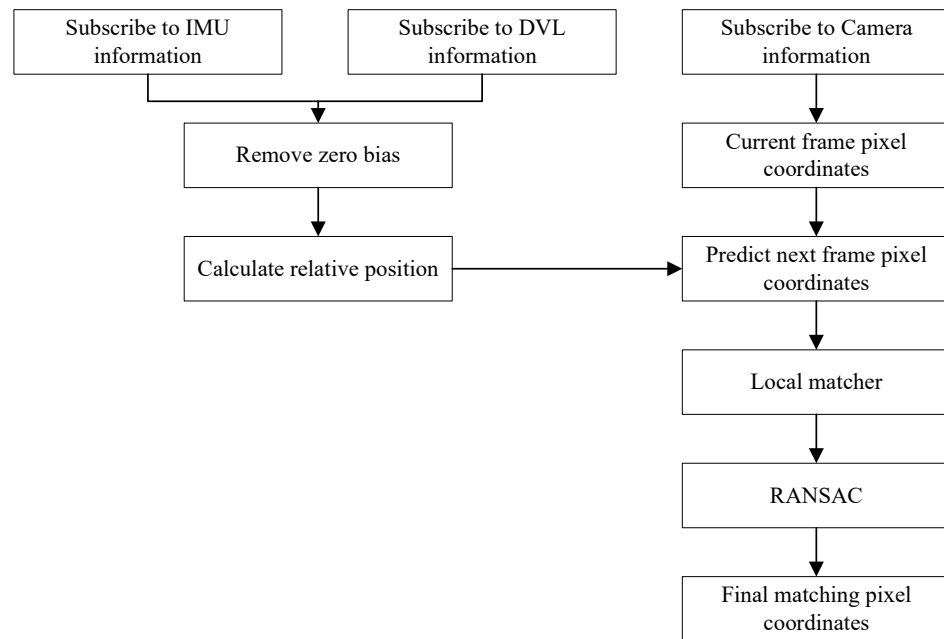


Figure 6. Underwater feature-matching flowchart.

3. Results

In this study, we verified the localization accuracy and robustness of the proposed algorithm through simulation and physical experiments, using the proposed algorithm as the experimental group and VINS-Mono as the control group.

3.1. Simulation Experiments

In this study, the underwater AFRL dataset [35] was selected for simulation verification. According to the authors’ suggestion, COLMAP [36] was used to generate the reference trajectories of the dataset, which could be considered as the ground truth (GT). To be more representative, the Bus subdataset with poorer image quality in the AFRL dataset and the Cave subdataset with better image quality were selected for simulation experiments. Only the “left eye” information was used during the simulation process. Additionally, new datasets including the DVL information were generated based on the two subdatasets according to Figure 7 to validate the SLAM algorithm proposed in this paper.

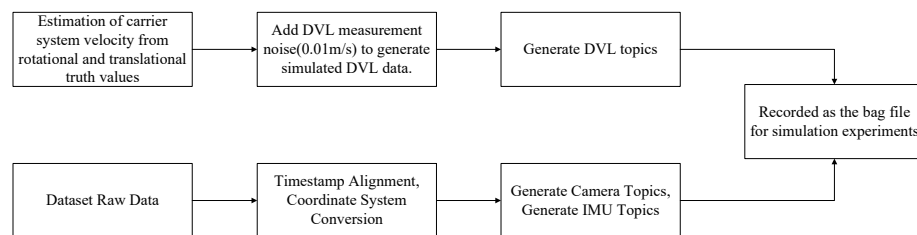


Figure 7. Underwater feature-matching flowchart.

The simulation experiment results are shown in Figures 8 and 9, and the experimental errors, including mean error (MEAN), standard deviation (STD), and root mean square error (RMSE), are shown in Table 1. From Figures 8 and 9, it can be seen that the VINS-Mono algorithm can run normally only on the Cave subdataset, but the absolute pose error (APE) is relatively large. This includes max (maximum error), min (minimum error), std (same as STD), median, mean (same as MEAN), and mse (mean square error). The SLAM algorithm proposed in this paper ran normally on both underwater subdatasets. Although the APE was a little larger on the Bus subdataset than on the Cave subdataset, it was still within a reasonable range.

From Table 1, it can be seen that compared with the VINS-Mono algorithm, the SLAM algorithm proposed in this paper has better adaptability and robustness. On the Bus subdataset, with a total mileage of about 53 m, the VINS-Mono algorithm could not run properly, while the RMSE of the SLAM algorithm proposed in this paper was 0.29 m. On the Cave subdataset, with a total mileage of about 87 m, the RMSE of the VINS-Mono algorithm was 0.35 m, while the RMSE of the SLAM algorithm proposed in this paper was 0.16 m, which is 54.29% lower.

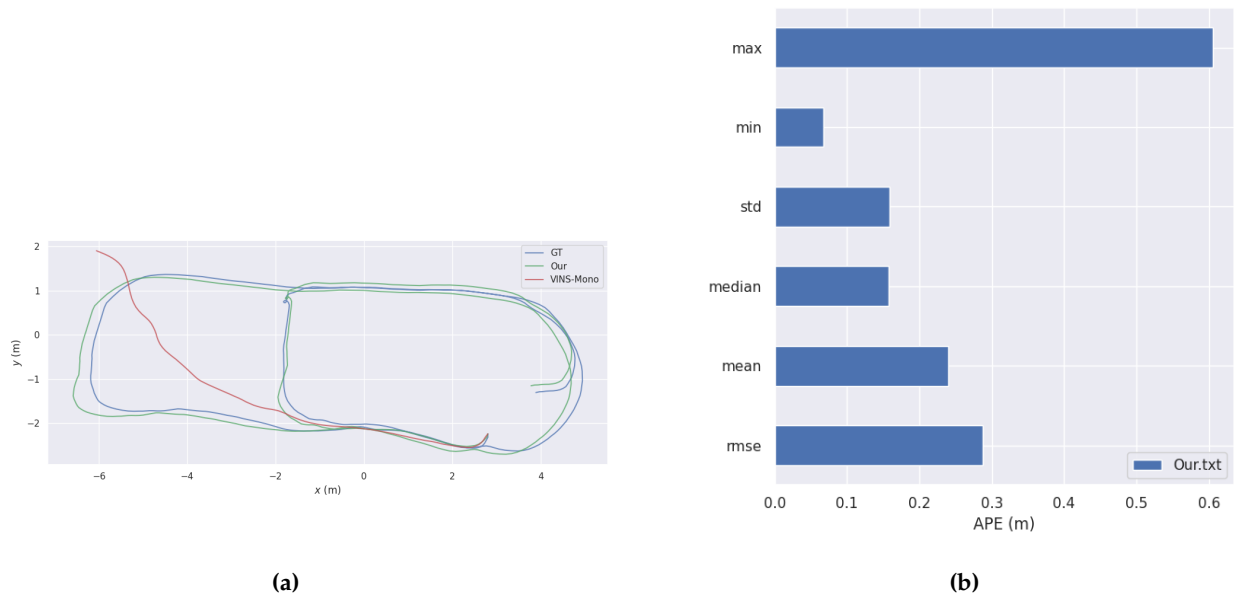


Figure 8. Comparison plot of the two algorithms on the Bus subdataset: (a) trajectory comparison chart; (b) APE comparison chart.

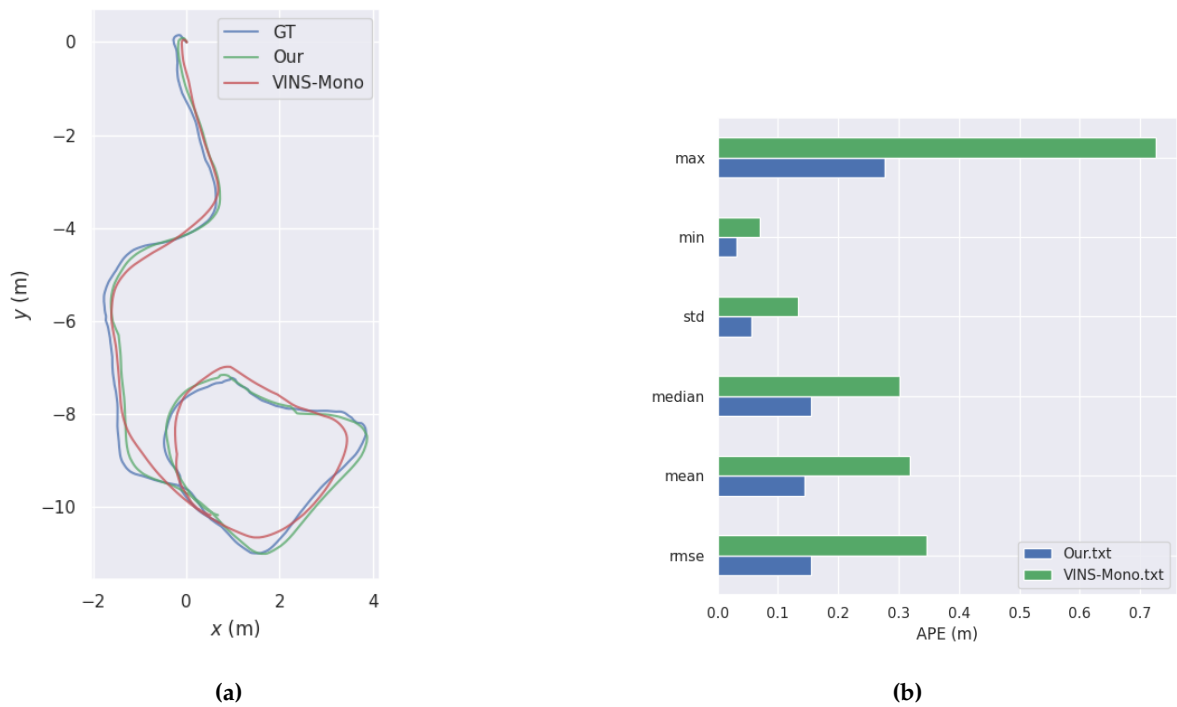


Figure 9. Comparison plot of the two algorithms on the Cave subdataset: (a) trajectory comparison chart; (b) APE comparison chart.

Table 1. Comparison of the errors of the two algorithms.

Subdataset	Algorithm	MEAN (m)	STD (m)	RMSE (m)
Bus subdataset	VINS-Mono *	-	-	-
	Our	0.24	0.16	0.29
Cave subdataset	VINS-Mono	0.32	0.13	0.35
	Our	0.14	0.06	0.16

* Indicates algorithm could not run properly.

3.2. Physical Experiments

In this study, the algorithm proposed in this paper was validated through underwater physical experiments. BlueROV was selected as the experimental platform, carrying sensors such as an underwater camera, an IMU, a DVL, and an underwater global positioning system (GPS). The underwater experiments were divided into open-loop and closed-loop experiments, and the reference trajectories (i.e., GT) were provided by the underwater GPS.

3.2.1. Open-Loop Experiment

The total length of the underwater open-loop experimental trajectory was about 83.18 m. The experimental trajectory is shown in Figure 10. RVIZ is a 3D visualizer for the Robot Operating System (ROS) framework.

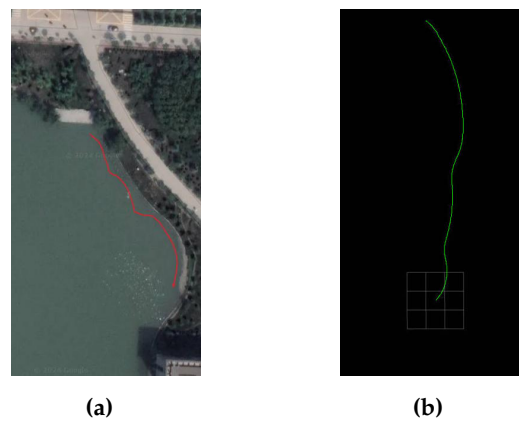


Figure 10. Underwater open-loop experimental trajectory: (a) experimental satellite trajectory map (hand-painted); (b) RVIZ trajectory map.

The experimental results are shown in Figures 11 and 12, and the experimental errors are shown in Table 2. From Figures 11 and 12, it can be observed that in the feature-rich region, both the proposed algorithm and VINS-Mono could detect the feature points that satisfy the operation of their respective algorithms. However, the proposed image enhancement algorithm significantly increased the number of feature points, which improved localization accuracy to a certain extent. From Table 2, in terms of mean absolute error (MAE) and STD, the proposed algorithm reduced these values by 1.20 m and 0.40 m, respectively, compared to those of the VINS-Mono algorithm.

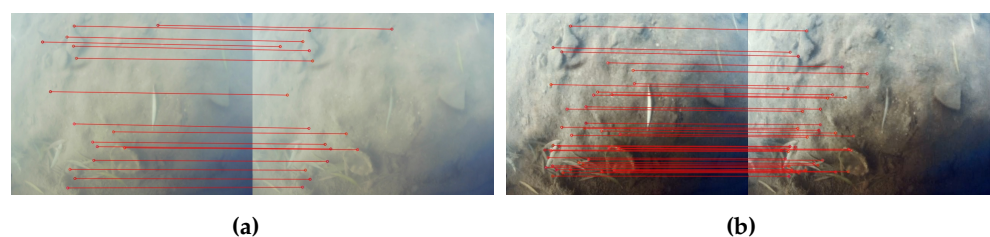


Figure 11. Two algorithms' feature-matching results: (a) VINS-Mono; (b) the algorithm proposed in this paper.

Table 2. Comparison of the errors of the two algorithms.

Algorithm	MAE (m)	STD (m)
VINS-Mono	1.76	0.90
Our	0.56	0.50

Building on VINS-Mono, the proposed algorithm uses image enhancement to optimize the original image. Additionally, the DVL used for velocimetry has a certain inhibitory effect on the velocity dispersion of the IMU accelerometer, resulting in better performance.

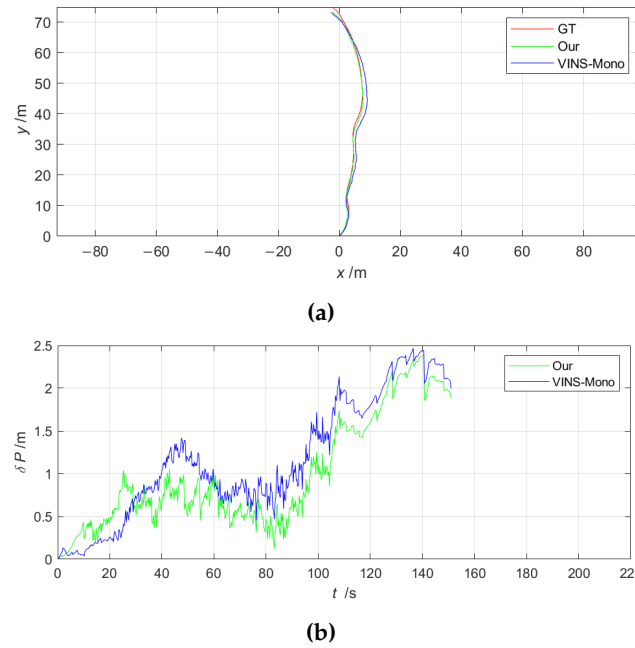


Figure 12. Results of underwater open-loop experiments with two algorithms: (a) trajectory comparison chart; (b) error comparison chart.

3.2.2. Closed-Loop Experiment

The total mileage of the underwater closed-loop experimental trajectory was about 209.04 m. The experimental trajectory is shown in Figure 13.

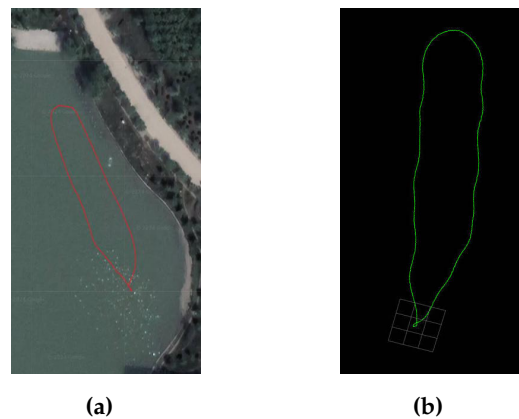


Figure 13. Underwater closed-loop experimental trajectory: (a) experimental satellite trajectory map (hand-painted); (b) RVIZ trajectory map.

The experimental results are shown in Figures 14 and 15, and the experimental errors are shown in Table 3. From Figures 14 and 15, it can be seen that in feature-scarce regions, VINS-Mono could not run properly due to its inability to detect the required feature points.

The image enhancement algorithm proposed in this paper effectively solved this problem by providing enough feature points for the algorithm, thereby guaranteeing its normal operation. From Table 3, in terms of MAE and STD, the proposed algorithm had errors of 0.84 m and 0.48 m, respectively, and these errors are relatively small. (Note: The errors of the VINS-Mono algorithm shown in Figure 15 and Table 3 represent the errors before it failed to run.)

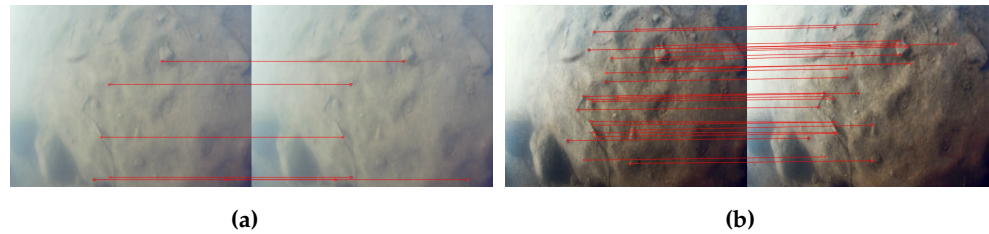


Figure 14. Two algorithms' feature-matching results. (a) VINS-Mono; (b) the algorithm in this paper.

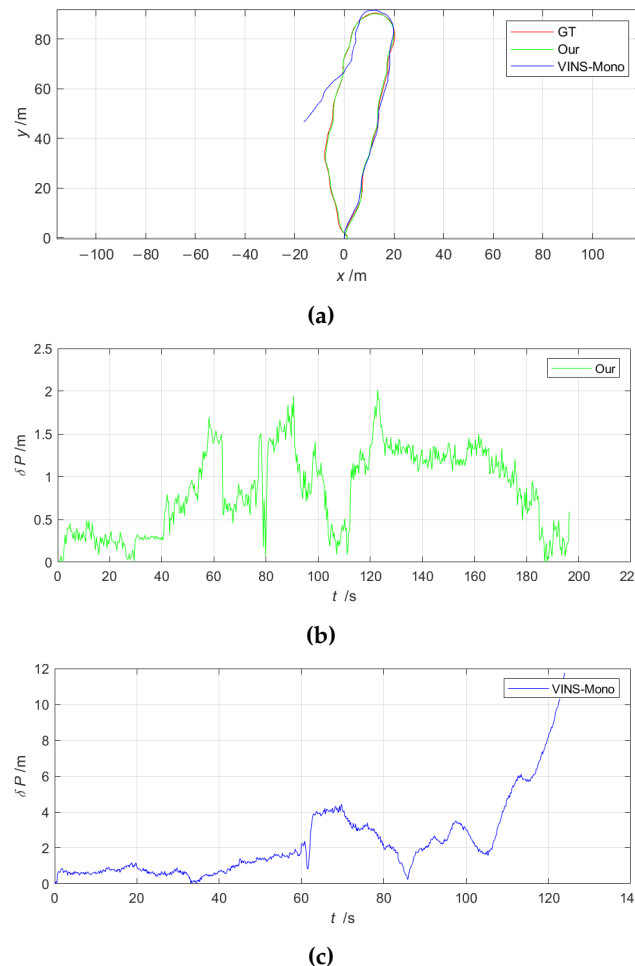


Figure 15. Results of underwater closed-loop experiments with two algorithms: (a) trajectory comparison chart; (b) error chart of the proposed algorithm; (c) error chart for the nonfailed part of the VINS-Mono algorithm. (Due to the significant numerical difference between the errors of the proposed algorithm and the VINS-Mono algorithm, the errors are plotted separately to better illustrate the details.)

Table 3. Comparison of the errors of the two algorithms.

Algorithm	MAE (m)	STD (m)
VINS-Mono *	2.32	2.01
Our	0.84	0.48

* Represents the error of the nonfailed part of the VINS-Mono algorithm.

In the underwater closed-loop experiments, the proposed algorithm detected the loopback and initiated the repositioning and position optimization session, resulting in a small error near the starting point. Additionally, the proposed algorithm incorporated more image enhancement and DVL constraints compared to the VINS-Mono algorithm, which improved robustness and positioning accuracy.

4. Discussion

To improve the accuracy and robustness of underwater visual SLAM, this paper proposes an underwater multisensor fusion SLAM system based on image enhancement. The system uses an underwater image enhancement algorithm based on a generative adversarial network to optimize the underwater images. An underwater feature-matching algorithm based on the local matcher is used to solve the feature-matching problem due to the grayscale change in the enhanced images. Additionally, the DVL is tightly coupled to the visual SLAM system, which ultimately reduces the accuracy error.

In the experiments, the main reason for the failure of the VINS-Mono algorithm was the low input image quality, which could not provide sufficient feature points. In contrast, the image enhancement part of the proposed algorithm effectively solves this problem. Additionally, the above optimization further improves the navigation accuracy of the proposed algorithm. In the open-loop experiment, the MAE and STD of the proposed algorithm were reduced by 68.18% and 44.44%, respectively, compared to those of VINS-Mono. In the closed-loop experiment, the MAE and STD values of the proposed algorithm were 0.84 m and 0.48 m, respectively.

The proposed algorithm is mainly suitable for areas with rich and easily distinguishable underwater features. However, in special underwater environments, AUVs sometimes move to areas with weak or repetitive textures, such as a flat seabed with all sand, resulting in insufficient or mismatched features. In this case, the proposed algorithm has the possibility of crashing. To solve this problem, in future research, we plan to add Kalman filtering to the algorithm as an alternative navigation scheme. We are investigating the navigation accuracy confidence between the two, the timing of switching between them, and the related problems faced by visual SLAM when rerunning on an existing map. Additionally, to further bolster the algorithm's robustness, we will explore integrating a magnetometer and a depth gauge into the algorithm, which will help with obtaining more accurate heading angle and depth information. Therefore, the code is not open-source.

5. Conclusions

In this paper, we proposed an underwater multisensor fusion SLAM system based on image enhancement, which uses a generative adversarial network with hybrid attention to enhance underwater images. To obtain feature matching between the enhanced images, based on the a priori information provided by the gyroscope and the DVL, we use the proposed local matcher to accomplish feature tracking. Meanwhile, the addition of the DVL information enhances the system's robustness and accuracy.

Specifically, the main focus of this study was on underwater image enhancement and tight DVL coupling. Underwater image enhancement is the foundation of the entire algorithm, directly determining whether the algorithm can run normally. Its role is to process poor-quality underwater images into images that meet the system's requirements, providing enough feature points for the algorithm. The role of tight DVL coupling is manifested in two aspects. First, it provides a priori information for the feature matching of the underwater enhanced image, increasing the speed and accuracy of the matching. Second,

it provides real speed information to optimize the initialization and nonlinear optimization of the algorithm, which in turn improve the accuracy of navigation and positioning.

To validate the proposed algorithm, we conducted AFRL dataset simulation experiments and physical experiments. The results of these experiments demonstrated that our proposed algorithm has good accuracy and underwater adaptability. Additionally, our system provides significant improvements in terms of MAE and STD over VINS-Mono.

Author Contributions: Conceptualization, Z.L. and F.Z.; methodology, Z.L.; software, Z.L.; validation, K.W. and F.Z.; formal analysis, Z.L., K.W., J.Z. and F.Z.; writing—original draft preparation, Z.L.; writing—review and editing, Z.L., K.W., J.Z. and F.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available dataset AFRL was analyzed in this study. These data can be found here: <https://afrl.cse.sc.edu/afrl/resources/datasets/> (accessed on 12 April 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AUV	Autonomous Underwater Vehicle
SLAM	Simultaneous Localization and Mapping
DVL	Doppler Velocity Log
MAE	Mean Absolute Error
STD	Standard Deviation
MEMS IMU	Micro-Electro-Mechanical System Inertial Measurement Unit
EKF	Extended Kalman Filter
PF	Particle Filter
VIO	Visual Inertial Odometer
GT	Ground Truth
MEAN	Mean Error
RMSE	Root Mean Square Error
APE	Absolute Pose Error
GPS	Global Positioning System
ROS	Robot Operating System
MAE	Mean Absolute Error

References

- Li, Y.; Takahashi, S.; Serikawa, S. Cognitive ocean of things: A comprehensive review and future trends. *Wirel. Netw.* **2022**, *28*, 917. [[CrossRef](#)]
- Lusty, P.A.; Murton, B.J. Deep-ocean mineral deposits: Metal resources and windows into earth processes. *Elements* **2018**, *14*, 301–306. [[CrossRef](#)]
- Constable, S.; Kowalczyk, P.; Bloomer, S. Measuring marine self-potential using an autonomous underwater vehicle. *Geophys. J. Int.* **2018**, *215*, 49–60. [[CrossRef](#)]
- Wu, Y. Coordinated path planning for an unmanned aerial-aquatic vehicle (UAAV) and an autonomous underwater vehicle (AUV) in an underwater target strike mission. *Ocean. Eng.* **2019**, *182*, 162–173. [[CrossRef](#)]
- Johnson-Roberson, M.; Bryson, M.; Friedman, A.; Pizarro, O.; Troni, G.; Ozog, P.; Henderson, J.C. High-resolution underwater robotic vision-based mapping and three-dimensional reconstruction for archaeology. *J. Field Robot.* **2017**, *34*, 625–643. [[CrossRef](#)]
- Mogstad, A.A.; Ødegård, Ø.; Nornes, S.M.; Ludvigsen, M.; Johnsen, G.; Sørensen, A.J.; Berge, J. Mapping the historical shipwreck *figaro* in the high arctic using underwater sensor-carrying robots. *Remote Sens.* **2020**, *12*, 997. [[CrossRef](#)]
- Ma, L.; Gulliver, T.A.; Zhao, A.; Zeng, C.; Wang, K. An underwater bistatic positioning system based on an acoustic vector sensor and experimental investigation. *Appl. Acoust.* **2021**, *171*, 107558. [[CrossRef](#)]
- Zhao, S.; Wang, Z.; Nie, Z.; He, K.; Ding, N. Investigation on total adjustment of the transducer and seafloor transponder for GNSS/Acoustic precise underwater point positioning. *Ocean. Eng.* **2021**, *221*, 108533. [[CrossRef](#)]

9. Hsu, H.Y.; Toda, Y.; Yamashita, K.; Watanabe, K.; Sasano, M.; Okamoto, A.; Inaba, S.; Minami, M. Stereo-vision-based AUV navigation system for resetting the inertial navigation system error. *Artif. Life Robot.* **2022**, *27*, 165–178. [[CrossRef](#)]
10. Mu, X.; He, B.; Wu, S.; Zhang, X.; Song, Y.; Yan, T. A practical INS/GPS/DVL/PS integrated navigation algorithm and its application on Autonomous Underwater Vehicle. *Appl. Ocean. Res.* **2021**, *106*, 102441. [[CrossRef](#)]
11. Chen, H.; Huang, H.; Qin, Y.; Li, Y.; Liu, Y. Vision and laser fused SLAM in indoor environments with multi-robot system. *Assem. Autom.* **2019**, *39*, 297–307. [[CrossRef](#)]
12. Zhao, J.; Liu, S.; Li, J. Research and implementation of autonomous navigation for mobile robots based on SLAM algorithm under ROS. *Sensors* **2022**, *22*, 4172. [[CrossRef](#)] [[PubMed](#)]
13. Peng, H.; Zhao, Z.; Wang, L. A Review of Dynamic Object Filtering in SLAM Based on 3D LiDAR. *Sensors* **2024**, *24*, 645. [[CrossRef](#)] [[PubMed](#)]
14. Huang, B.; Zhao, J.; Liu, J. A survey of simultaneous localization and mapping with an envision in 6g wireless networks. *arXiv* **2019**, arXiv:1909.05214.
15. Shangguan, M.; Weng, Z.; Lin, Z.; Lee, Z.; Shangguan, M.; Yang, Z.; Sun, J.; Wu, T.; Zhang, Y.; Wen, C. Day and night continuous high-resolution shallow-water depth detection with single-photon underwater lidar. *Opt. Express* **2023**, *31*, 43950–43962. [[CrossRef](#)] [[PubMed](#)]
16. Shangguan, M.; Yang, Z.; Lin, Z.; Lee, Z.; Xia, H.; Weng, Z. Compact Long-Range Single-Photon Underwater Lidar with High Spatial–Temporal Resolution. *IEEE Geosci. Remote. Sens. Lett.* **2023**, *20*, 1501905. [[CrossRef](#)]
17. Macario Barros, A.; Michel, M.; Moline, Y.; Corre, G.; Carrel, F. A comprehensive survey of visual slam algorithms. *Robotics* **2022**, *11*, 24. [[CrossRef](#)]
18. Chen, W.; Shang, G.; Ji, A.; Zhou, C.; Wang, X.; Xu, C.; Li, Z.; Hu, K. An overview on visual slam: From tradition to semantic. *Remote Sens.* **2022**, *14*, 3010. [[CrossRef](#)]
19. Tena Ruiz, I.J. Enhanced Concurrent Mapping and Localisation Using Forward-Looking Sonar. Ph.D. Thesis, Heriot-Watt University, Edinburgh, UK, 2001.
20. Choi, J.; Lee, Y.; Kim, T.; Jung, J.; Choi, H.T. EKF SLAM using acoustic sources for autonomous underwater vehicle equipped with two hydrophones. In Proceedings of the OCEANS 2016 MTS/IEEE Monterey, Monterey, CA, USA, 19–23 September 2016; pp. 1–4.
21. Wang, W.; Cheng, B. Augmented EKF based SLAM system with a side scan sonar. In Proceedings of the 2020 12th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), Hangzhou, China, 26–27 August 2020; IEEE: New York, NY, USA, 2020; Volume 1, pp. 71–74.
22. Chen, L.; Yang, A.; Hu, H.; Naeem, W. RBPF-MSIS: Toward rao-blackwellized particle filter SLAM for autonomous underwater vehicle with slow mechanical scanning imaging sonar. *IEEE Syst. J.* **2019**, *14*, 3301–3312. [[CrossRef](#)]
23. Zhang, Q.; Li, Y.; Ma, T.; Cong, Z.; Zhang, W. Bathymetric particle filter SLAM based on mean trajectory map representation. *IEEE Access* **2021**, *9*, 71725–71736. [[CrossRef](#)]
24. Chen, F.; Zhang, B.; Zhao, Q. Multi-AUVs Cooperative SLAM Under Weak Communication. In Proceedings of the 2023 International Conference on Control, Robotics and Informatics (ICCRI), Shanghai, China, 26–28 May 2023; IEEE: New York, NY, USA, 2023; pp. 52–56.
25. Joshi, B.; Bandara, C.; Poulakakis, I.; Tanner, H.G.; Rekleitis, I. Hybrid Visual Inertial Odometry for Robust Underwater Estimation. In Proceedings of the OCEANS 2023-MTS/IEEE US Gulf Coast, Biloxi, MS, USA, 25–28 September 2023; IEEE: New York, NY, USA, 2023; pp. 1–7.
26. Yang, Y.F.; Qin, J.H.; Li, T. Study on the light scattering of suspended particles in seawater. *J. Electron. Meas. Instrum.* **2018**, *32*, 145–150.
27. Zhai, C.C.; Han, X.Y.; Peng, Y.F. Research on light transmission characteristics of some inorganic salts in seawater. *Laser Optoelectron. Prog.* **2014**, *52*, 43–48.
28. Qin, T.; Li, P.; Shen, S. VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. *IEEE Trans. Robot.* **2018**, *34*, 1004–1020. [[CrossRef](#)]
29. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *Commun. ACM* **2020**, *63*, 139–144. [[CrossRef](#)]
30. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, 5–9 October 2015; Proceedings, Part III; Springer International Publishing: Cham, Switzerland, 2015; pp. 234–241.
31. Zhao, H.; Gallo, O.; Frosio, I.; Kautz, J. Loss functions for image restoration with neural networks. *IEEE Trans. Comput. Imaging* **2016**, *3*, 47–57. [[CrossRef](#)]
32. Forster, C.; Carlone, L.; Dellaert, F.; Scaramuzza, D. On-manifold preintegration for real-time visual–inertial odometry. *IEEE Trans. Robot.* **2016**, *33*, 1–21. [[CrossRef](#)]
33. Tamgade, S.N.; Bora, V.R. Notice of Violation of IEEE Publication Principles: Motion Vector Estimation of Video Image by Pyramidal Implementation of Lucas Kanade Optical Flow. In Proceedings of the 2009 Second International Conference on Emerging Trends in Engineering & Technology, Nagpur, India, 16–18 December 2009; IEEE: New York, NY, USA, 2009; pp. 914–917.
34. Fischler, M.A.; Bolles, R.C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **1981**, *24*, 381–395. [[CrossRef](#)]

35. Rahman, S.; Quattrini Li, A.; Rekleitis, I. SVIn2: A multi-sensor fusion-based underwater SLAM system. *Int. J. Robot. Res.* **2022**, *41*, 1022–1042. [[CrossRef](#)]
36. Schonberger, J.L.; Frahm, J.M. Structure-from-Motion Revisited. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; IEEE: New York, NY, USA, 2016; pp. 4104–4113.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.