Journal of Electronics and Information Science (2025)
Clausius Scientific Press, Canada

DOI: 10.23977/jeis.2025.100101
ISSN 2371-9524 Vol. 10 Num. 1

# *Underwater Monocular-continuous Stereo Network Based on Cascade Structure for Underwater Image Depth Estimation*

**Yao Haiyang[1,*], Zeng Yiwen[1], Zang Yuzhang[2], Lei Tao[1], Zhao Xiaobo[3], Chen Xiao[1], Wang Haiyan[1,4]**

*[1]School of Electronic Information and Artificial Intelligence, Shaanxi University of Science and Technology, Xi'an, 710016, China*
*[2]Engineering and Design Department, Western Washington University, Bellingham, WA, USA*
*[3]Department of Electrical and Computer Engineering, Aarhus University, Aarhus, 8200, Denmark*
*[4]School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an, 710072, China*
*[*]Corresponding author: yaohy1991@126.com*

***Keywords:*** Underwater monocular images, underwater depth estimation, ocean engineering, deep learning

***Abstract:*** Underwater monocular image depth estimation (UMIDE) is crucial accurately representing and understanding underwater spatial variations, which can significantly enhance applications such as ocean engineering construction and seabed resource exploration. However, UMIDE frequently suffers from isolated discontinuous irregular "spots", inaccurate or indistinguishable edges, and limited model generalization, resulting from color distortion, image blurring, and spatial information loss. This paper proposes an underwater Monocular-continuous stereo network based on a cascade structure (UMCS-CS). Initially, we design a Pinhole model-based Structure from Motion method for camera pose estimation. UMCS-CS employs a two-stage structure for feature extraction: the first stage extracts global information, and the second stage captures detailed information using the squeeze–excitation block with spatial and channel attention. For isolated, discontinuous, and irregular "spots", we use the variance of the current depth estimation to adjust and appropriately expand the depth estimation range. We design a composite loss function, which is a combination of the smooth L1 loss, edge loss function, structural similarity loss, and smoothness loss functions, each with different weights. Experiments on public underwater datasets show that the relative error of the estimated depth map is reduced by 60.83%, the root mean square error by 54.87%, and the logarithmic error by 39.61%.

## 1. Introduction

The complex marine environment poses significant challenges for underwater resource exploration and ocean engineering construction. Automatic and intelligent equipment play a pivotal role in future underwater missions. Monocular imaging equipment is suitable for nearly all underwater vehicles and has become a critical component for underwater unmanned systems to perceive their environment, especially for small equipment performing special tasks. Depth prediction for underwater optical images is a critical technology in 3D reconstruction, with significant applications in marine engineering, resource exploration, and future seabed infrastructure development. Given the highly complex physical environments of underwater spaces, autonomous underwater vehicles (AUVs) are the sole viable platforms for task execution. Monocular optical imaging is a crucial sensory modality for small AUVs, underscoring the importance of depth prediction for monocular optical images. Underwater image monocular depth estimation (UMIDE) is a key technology for perceiving complex seabed environments and understanding spatial variations [1]. It provides essential three-dimensional information for seabed topography mapping, crucial for underwater resource exploration and precise object detection [2]. Additionally, UMIDE informs excavation depths and ranges for submarine pipelines and the placement of large equipment [3].

However, the depth information is largely obscured in monocular vision imaging, and the absence of images from different positions in monocular images exacerbates the challenge. Moreover, the complex underwater environment affects light absorption, scattering, and refraction, resulting in color distortion, reduced contrast, and image blurring [4]. Furthermore, underwater monocular image datasets with accurate environmental depth labels and camera pose information are scarce [5].

This paper proposes an underwater Monocular-continuous stereo network based on a cascade structure for UMIDE. Our study addresses three primary challenges in underwater monocular image depth prediction: discontinuous irregular "spots," inaccurate or indistinguishable edges, and limited model generalization. Through in-depth research into topics such as underwater optical image enhancement and target detection, we identified theoretical and practical issues in depth prediction. Extensive experiments with various depth prediction methods for underwater monocular optical images allowed us to refine and clearly define these challenges through detailed comparative analysis.

## 2. Literature review

Monocular image depth estimation, which involves predicting the depth of a scene from a single or a sequence of RGB images captured from a fixed viewpoint, has emerged as a significant research topic in computer vision. In underwater scenarios, monocular image sequences estimate depth by employing triangulation technology to calculate parallax; however, matching is significantly affected by the angle of illumination. Traditional methods focus on developing mathematical models or fitting techniques to define the mapping relationship between RGB images and depth maps, while deep learning approaches aim to learn this mapping function using various neural networks architectures.

### 2.1 Traditional methods for underwater monocular image depth estimation

The optical transmission mode serves as a critical foundation and prior knowledge for depth estimation tasks. Song et al. introduced a fast and effective depth estimation model based on Underwater Light Attenuation Prior (ULAP), defining the difference between the maximum G-B intensity (MVGB) and the value intensity [6]. Berman et al. proposed a method for optical transmission estimation consisting of three steps: blocked light estimation (accounting for ambient light scattered into the line of vision), transmission estimation for different water types with varying optical properties, and automatic result selection based on the grey world hypothesis [7]. Chen et al. improved the speed of depth estimation by optimizing the selection of key frames; they extracted frames from video streams as SfM image sequences and selected global SfM to construct scenes [8]. Schonberger et al. developed the COLLISION-Mapping (COLMAP) system, which further integrates normal vector information of pixels to enhance the quality of depth estimation [9]. However, the performance of traditional methods is limited in areas with weak textures. To address this, Romanoni and Matteucci proposed the TAPA-MVS method, which assumes segmented plane characteristics in weak texture regions and uses this assumption as prior knowledge to guide the COLMAP system in depth estimation for these areas [10]. Zhu et al. refined the process of mark matching by combining multi-shot fusion perceptual prediction with K-Means clustering algorithm [11]. However, traditional methods generally suffer from limited robustness and generalization capabilities. Depth estimation accuracy tends to degrade in low-texture scenes, under poor lighting conditions, or when applied to non-reference scenarios.

## 2.2 Deep Learning Based Methods for Underwater Monocular Image Depth Estimation

Recent advancements in deep learning have introduced innovative approaches to underwater depth estimation, leveraging neural networks to handle large-scale data and diverse underwater scenes and objects. Levy et al. designed an architecture capable of learning scene information and medium parameters to eliminate the medium between the camera and the scene, thereby reconstructing the appearance and depth of distant objects [12]. Ye et al. proposed an underwater depth estimation network that adaptively infers depth maps from underwater stereoscopic images, utilizing three different adaptation modules: style adaptation, semantic adaptation, and parallax range adaptation [13]. Li et al. designed a multi-stage and multi-task learning framework for predicting dense relative depth map of underwater monocular image [14]. Gupta et al. employed a dense block-based autoencoder as a generator network to learn the mapping function between unpaired RGB-D ground images and arbitrary underwater images, estimating the desired depth map [15]. Nagamatsu et al. proposed a self-calibration method using visual ranging to bundle numerous frames, thereby increasing density in 3D reconstruction of monocular images [16]. Zhang et al. introduced a geometric perception model called GeoMVSNet, which integrates geometric cues present in the rough stage of processing [17]. Qi et al. proposed a geometric neural network, GeoNet++, featuring edge perception refinement to jointly estimate depth and surface normal of the monocular images through depth-to-normal and normal-to-depth modules [18]. Marques developed a framework capable of real-time depth estimation of underwater images while also estimating the model's uncertainty in predicting the depth [19]. Li et al. utilized a stereo matching method based on the optimal search domain to enhance stereo matching accuracy in underwater scenes [20]. Ebner et al. incorporated sparse depth measurements from triangulated features to improve depth estimation and address scaling ambiguity [21]. Finally, Ye et al. proposed an unsupervised adaptation network to learn domain-invariant representations, enabling joint estimate of scene depth and color correction from underwater monocular images [22].

Deep learning methods excel in handling large-scale data and accommodating diverse types of scenes and objects. However, existing deep learning methods still encounter several challenges in depth estimation:

1) Discontinuous 'spots' appear in the depth prediction results, occurring irregularly across different positions in the monocular image, with no clear pattern or regularity.

2) Errors frequently occur in the depth prediction along the edge between objects and their background, especially when the edges are not smooth, or when the contrast between the edge and the background is minimal. These issues not only produce inaccuracies but also render the edges indistinguishable.

3) The generalization capability of the model enhancement, as the depth prediction results across different datasets still require further improvement.

We address the aforementioned issues on two levels. At the level of underwater optical image feature representation, we enhance the prior knowledge of optical images by integrating structural models and matching techniques to mitigate the irregular disturbances caused by the underwater physical environment during depth prediction. To reduce inaccuracies in edge depth prediction, we combine local and global features, enabling the capture of a more comprehensive range of colour variations within the images. At the model structure level, we design a cascade architecture to extract multi-layer variation information from optical images. Additionally, we propose a composite loss function to enhance the model's generalization ability. The following sections provide a detailed description on the specific structure of the UMCS-CS network.

## 3. Methodology

The proposed Underwater Monocular-Continuous Stereo Network (UMCS-CS) is a deep learning framework designed to enhance underwater depth estimation by leveraging a cascade structure and a composite loss function. This section details the network architecture, feature extraction process, and loss function design that together address the challenges of depth estimation in underwater environments.

### 3.1 Network architecture

The UMCS-CS network is structured as a convolutional neural network specifically designed for underwater depth estimation, following a series of methodical steps as illustrated in Figure 1. The network begins by extracting camera pose information from underwater optical images and constructs a cost volume with regularization for depth estimation from underwater monocular optical image sequences. The feature extraction process is divided into two stages.

Employing a cascade structure, the network starts with feature maps at a smaller scale and performs homograph transformations on feature maps of adjacent frames to create a feature volume, which is then used to construct a cost volume. After normalizing the regularized cost volume, the network generates an estimated depth map. This depth map is used to progressively refine the depth intervals at the next scale in a coarse-to-fine manner [23], enabling hierarchical inference of depth maps for the reference images.

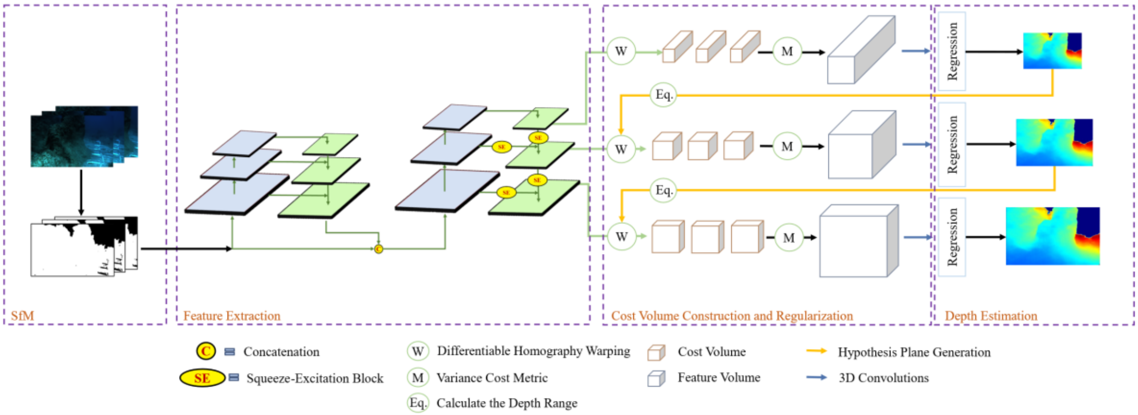
SfM
Feature Extraction
Cost Volume Construction and Regularization
Depth Estimation
W
M
Eq.
SE
C
Regression
Regression
Regression
C Concatenation
SE Squeeze-Excitation Block
W Differentiable Homography Warping
M Variance Cost Metric
Eq. Calculate the Depth Range
Cost Volume
Feature Volume
Hypothesis Plane Generation
3D Convolutions

Figure 1: The network design of UMCS-CS

### 3.2 Pinhole model-based Structure from Motion method

The image features are extracted and matched using the Pinhole model [24], and the camera parameters along with the corresponding rough depth range are determined through data format conversion. Masks are generated based on provided depth maps and original images. Regions where depth values are undefined in the ground truth depth maps are considered invalid, while regions with defined values are valid. Mask1 is created based on this characteristic. Since the dataset pertains to shallow water areas prone to light spots, the images are converted from the RGB color space to the HIS color space. Regions where the saturation (S) is less than α1 and intensity (I) is greater than α2 are identified as high-brightness pixels. Each high-brightness pixel is expanded to its surrounding five pixels to define regions with light spots, forming mask2. Specifically, mask2 was designed to filter out irregular bright spots caused by lighting conditions. By applying mask2 to filter these regions, we effectively eliminate the influence of irregular bright spots during subsequent processing, thereby reducing the presence of irregular spots in the images. The final mask is obtained by merging mask1 and mask2. (Figure 2)

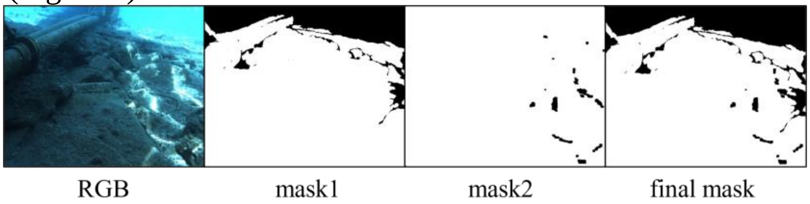

Figure 2: Illustration of the created masks: mask1 filters out invalid depth regions, and mask2 filters out glare regions

### 3.3 Two stage encoder-decoder feature extraction

Due to light absorption and refraction in water, underwater optical images often exhibit blurred textures and edges, leading to poor performance in the fine details of the estimated depth map. To address this issue, we propose a two-stage approach for feature extraction. The first stage focus on extracting global information, while the second stage integrates local information with the input image and incorporates SE(Squeeze-Excitation) blocks with spatial and channel attention to generate more refined feature maps.

As illustrated in Figure 1 and Figure 2, the feature extraction component consists of two pyramid structures representing the first stage. The input underwater reference images and their corresponding source images undergo separate feature extraction process. The input underwater image $p(u,v) \in P \subset \mathbb{R}^{H*W*3}$ is first processed through a feature encoder to obtain feature response $y \in Y \subset \mathbb{R}^{H/2k*W/2k*lc}$, then through a decoder to capture global information $x \in X \subset \mathbb{R}^{H*W*l}$. Here, H and W are the resolutions of the input images, k is a hyperparameter controlling the feature resolution, and lc denotes the number of channels in the output of the l-th layer. The number of layers in both the encoder and decoder pyramids are equal. Each encoder layer receives features from the preceding layer, and the output from the (L-l')-th layer of pyramid encoder is concatenated with the l'-th layer pyramid decoder to serve as input for the subsequent decoder layer.

The feature extraction in the second stage is composed of a two-layer pyramid structure. Building on the first-stage structure, we introduced our designed SE (Squeeze-and-Excitation) block into the skip connections and upsampling processes. The global information extracted in the first stage is concatenated with the input image to form a combined input $z \in Z \subset \mathbb{R}^{H*W*4}$, which is then fed into the second encoder. The global features extracted in the first stage are combined with the original image as a unified input for the second stage. After multiple layers of pyramid encoding, the upsampling output is processed through an SE block, while submodules in the encoder with the same resolution are also processed through SE blocks. The results of these two processes are concatenated as the input for the next submodule in the decoder. The SE block utilizes a compression and excitation mechanism to enhance the feature mappings of both local and global learning. The output from the (L−l)-th layer pyramid encoder, processed by an SE block, is concatenated with the output from the $l$-th layer pyramid decoder, also processed by an SE block. This concatenated result serves as the input for the next decoder layer. This mechanism enables the network to capture more detailed features and improves its multi-scale understanding of the image.

The primary function of SE blocks is to introduce an attention mechanism across the spatial and channel directions of the feature maps. These blocks are composed of SSCE (Spatially-Squeeze for Channel-wise Excitation) and CSSE (Channel-wise Squeeze for Spatially Excitation) blocks, as illustrated in Figure 3.

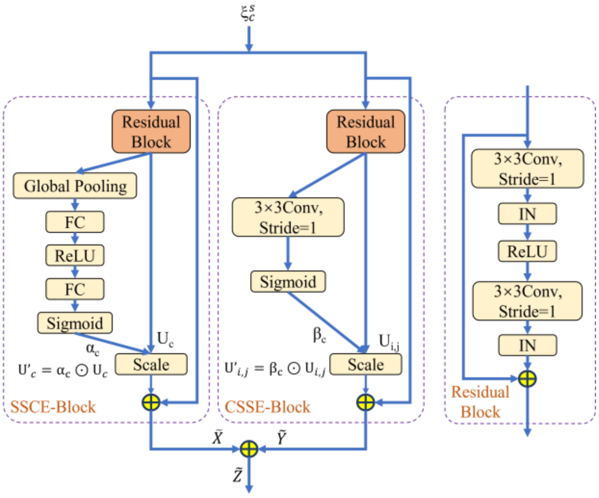

Figure 3: SE block structure

The feature map $\xi_c^s$ serves as the input for the SSCE block. Within the SSCE block, the input first undergoes processing through a residual block to produce the feature map $U_c$. Subsequently, global average pooling is applied to compresses the spatial direction of the feature map. The globally pooled features are then passed through fully connected layers and activation functions, where they are activated to generate attention weights $\alpha_c$ across channels. The resulting feature map $U'_c$ is calculated as:

$$U'_c = \alpha_c \odot U_c \tag{1}$$

where $\odot$ represents element-wise multiplication. The feature map $\xi_c^s$ is then element-wise summed with $U'_c$, followed by the application of a ReLU activation function to compress in spatial direction and excite in channel direction, resulting in the refined feature map $\tilde{X}$ :

$$\tilde{X} = relu(U'_c + \xi_c^s) \tag{2}$$

In the CSSE block, the feature map $U_{i,j}$ is first obtained after processing through a residual block. The channel direction is then compressed using a convolution operation, and then the convolved features are excited by an activation function to generate spatial attention weights $\beta_c$. The resulting feature map $U'_{i,j}$ is computed as:

$$U'_{i,j} = \beta_c \odot U_{i,j} \tag{3}$$

Next, the feature map $\tilde{Y}$ is obtained by element-wise summing $U'_{i,j}$ with $U_{i,j}$, followed by applying a ReLU activation function to compress in the channel direction and excite in the spatial direction, resulting in:

$$\tilde{Y} = relu\left(U'_c + \xi_c^s\right) \tag{4}$$

The final output of the SE block is:

$$\tilde{Z} = relu(\tilde{X} + \tilde{Y}) \tag{5}$$

The SE block thus enhances the generation of more refined feature maps by the second-stage decoder, enabling further effective extraction of underwater image features.

### 3.4 Cascade structure construction and regularization

To construct the cost volume, a cascade structure is employed. Initially, a differentiable homographic transformation is applied on a low-resolution feature map to map pixels from the source view to the reference view. The homography matrix is defined as:

$$H_i(d) = K_i \cdot R_i \cdot \left(I - \frac{(t_1 - t_i)n_1^T}{d}\right) \cdot R_1^T \cdot K_1^T \tag{6}$$

where I represents the identity matrix, $K_i$ and $K_1$ are the intrinsic parameter matrices of the source and reference cameras, respectively. Similarly, $R_i$ and $R_1$ denote the rotation matrices of the source and reference cameras, while $t_i$ and $t_1$ are the translation vectors of the source and reference cameras, respectively. In addition, d denotes the depth plane, and $n_1$ represents the principal axis of the reference camera.

Subsequently, at the l-th layer, the variance of features from N transformed source views and their corresponding features from the reference view at depth d is used to construct the cost volume $C_l$. A 3D convolutional network is then employed to regularize this cost volume and generate the probability volume $P_l$.

### 3.5 Composite loss function for depth estimation

The probability volume $P_l$ contains t pixel probability information across each depth plane. The depth value $d_l(p)$ for each pixel point p is estimated using:

$$d_l(p) = \sum_{n_d=0}^{N_d-1} d_{n_d} \times P_l(p) \tag{7}$$

where $d_{n_d}$ is the depth value of the $n_d$-th depth plane, and $P_l(p)$ represents the probability of pixel point p on that depth plane.

After obtaining the low-resolution depth map, bilinear interpolation is applied to upsample it. The depth estimation range of the previous layer is used as a reference to refine the current layer's depth estimation range and interval, resulting in more accurate depth estimation.

We designed a feedback mechanism between receptive fields of different sizes. This mechanism leverages the combination of small and large receptive fields to detect irregular variations across scales, enabling the network to better identify and distinguish irregular spots. As a result, the network achieves a more accurate representation of true object depths, mitigating the influence of these artifacts.

The depth plane d can be expressed as:

$$d = d_{min} + \frac{n_d(d_{min} - d_{max})}{N_d}, n_d \in \{0,1,\dots,N_d\} \tag{8}$$

where $d_{min}$ and $d_{max}$ represent the minimum and maximum values of the depth, respectively, and $N_d$ is the total number of depth planes sampled uniformly within the range from $d_{min}$ to $d_{max}$, and $n_d$ denotes the current number of depth planes.

To reduce errors caused by depth estimation inaccuracies, the variance of the current depth estimation is used to dynamically adjust the range, allowing for appropriate expansion. The new depth range is expressed as:

$$d'_{min} = max\{d_{min} - \frac{1}{H\times W}\sum_{u=0}^{H-1}\sum_{v=0}^{W-1}(d_{u,v}^{l} - \bar{d}^{l})^2\,, d_{min1}\} \tag{9}$$

$$d'_{max} = min\{d_{max} + \frac{1}{H\times W}\sum_{u=0}^{H-1}\sum_{v=0}^{W-1}(d_{u,v}^{l} - \bar{d}^{l})^2\,, d_{max1}\} \tag{10}$$

where $d'_{min}$ and $d'_{max}$ are the updated minimum and maximum depth values, $d_{min1}$ and $d_{max1}$ are the initial minimum and maximum depth values, respectively. $d_{i,j}^{l}$ is the depth value of pixel $p(u,v)$ at level l, and $\bar{d}^{l}$ is the average value of depth at level l. The new depth plane $d'$ can be expressed as:

$$d' = d'_{min} + \frac{n_d(d\prime_{min} - d\prime_{max})}{N_d}, n_d \in \{0,1,\dots,N_d\} \tag{11}$$

Once the new depth range and depth interval are obtained, the above process is repeated to perform depth map estimation at the next level.

At each level of estimated depth map, we use the Smooth L1 Loss (SLL), edge loss function, structural similarity loss function, and smoothness loss function to measure the discrepancy between the estimated depth map and the ground truth depth map. To focus on the valid regions and reduce the impact of speckle noise, we apply $M_i$, which denotes the mask of $F_i(p)$, considering only the areas filtered by the mask in the loss computation. We define:

$d_l^{GT}(p) = d_l^{GT}(p)[mask],\ d_l(p) = d_l(p)[mask],\ F(p) = F(p)[mask].$

where $d_l^{GT}(p), d_l(p)$ represent the depth value of point p in the ground truth depth map and estimated depth map at level l, respectively; and $F(p)$ represents the pixel value of point p.

The smooth L1 Loss Function measures the discrepancy between the estimated and ground truth depth values at each pixel. It balances robustness against outliers with precision and mitigates the risk of gradient explosion to a certain extent. The Loss$S$LL is expressed as:

$$Loss_{SLL} = \begin{cases} 0.5(d_l^{GT}(p) - d_l(p))^2, if\left|d_l^{GT}(p) - d_l(p)\right| < 1 \\ |d_l^{GT}(p) - d_l(p)| - 0.5, otherwise \end{cases} \tag{12}$$

The Edge Loss Function enhances the model's ability to extract spatial features from edges, this loss improves the accuracy of estimating high-frequency edge information, particularly at the boundaries between objects and scenes in the image. This function utilizes the Sobel operator to compute the gradients of both the estimated and ground truth depth maps. The loss is then calculated based on the difference between these two gradients. The calculation is given by:

$$Loss_{Edge} = \sum(\|d_l{}'(p) - d_l^{GT}{}'(p)\|)^2 \tag{13}$$

where $d_l{}'(p) = ((E_{g_1}^{H})^2 + (E_{g_1}^{V})^2)^{1/2}$ and $d_l^{GT}{}'(p) = ((E_{g_2}^{H})^2 + (E_{g_2}^{V})^2)^{1/2}$ , represent the gradients of the estimated and ground truth depth maps, respectively, as computed using horizontal and vertical gradient operators. $E^H$ and $E^V$ represent horizontal and vertical gradient operators.

From a global perspective, it is crucial to refine the visual perception system to accurately identify structural information from underwater scenes. This involves detecting differences between the reconstructed image and the reference scene from which the data is derived. The Structural Similarity Index (SSIM) is employed to assess the similarity of images by extracting brightness, contrast, and structural features. It measures the similarity of brightness $l(d_l(p), d_l^{GT}(p))$ , contrast $c(d_l(p), d_l^{GT}(p))$, and structure $s(d_l(p), d_l^{GT}(p))$ across different scales, ranging from local to global. The objective is to ensure that the predicted images are closely aligned with real images at a global scale. The SSIM loss function is:

$$Loss_{SSIM} = l\left(d_l(p), d_l^{GT}(p)\right) \cdot c\left(d_l(p), d_l^{GT}(p)\right) \cdot s\left(d_l(p), d_l^{GT}(p)\right) \tag{14}$$

We define the statistical measures employed for comparing distributions $d_l(p)$ and $d_l^{GT}(p)$.

Due to the presence of noise in underwater images, the estimated depth map may exhibit discontinuous. The Smoothness Loss Function assists the model in generating smoother depth maps that better align with the edges of the original image, ensuring consistency and realism. The formula for calculating the smoothness loss is given by:

$$L_{Soo} = \sum_p (\|\partial_x d_l(p)\| e^{-\|\partial_x F(p)\|} + \|\partial_y d_l(p)\| e^{-\|\partial_y F(p)\|}) \tag{15}$$

The total loss of the l-th level of UMCS-CS is defined as:

$$Loss_l = \alpha Loss_{SLL} + \beta Loss_{Edge} + \gamma Loss_{SSIM} + \zeta L_{Soo} \tag{16}$$

where α, β, $\gamma$ and $\zeta$ are the corresponding weights. Since depth maps generated at different levels have varying resolutions, and low-resolution depth maps assist in refining high-resolution ones, the contributions of depth maps at different resolutions should not be equally weighted during backpropagation. Consequently, when computing the overall network loss, the losses for each level are not directly summed. Instead, the loss for each level is weighted separately being combined. The overall loss is calculated as:

$$Loss = \sum_{l=1}^{3} \lambda_l \times Loss_l \tag{17}$$

where $Loss_l$ represents the loss generated by the estimated depth map at the l-th level, and $\lambda_l$ denotes the weight associated with loss at that level.

## 4. Experiments

In this section, we present experiments conducted on the datasets described in Section 4.1 to validate the effectiveness of the proposed method. A comprehensive evaluation is performed using both qualitative and quantitative approaches. The effectiveness of the method is demonstrated by quantitatively comparing the final predicted depth maps with the ground truth data. Additionally, comparisons with other learning-based depth estimation techniques highlight the superiority of the proposed framework.

### 4.1 Datasets and training process

The input data for the UMCS-CS network consists of six datasets:

1) BlendedMVS Dataset [25]: This dataset contains 113 scenes, each with 20 to 1000 images capturing various camera trajectories, amounting to a total of 17,818 images.

2) FLSea Dataset [26]: This dataset comprises 5056 pairs of depth maps and optical images collected in shallow waters (less than 10 meters) in the Mediterranean and Red Seas.

3) Torpedo Boat Wreck Dataset [27]: Filmed off the southern coast of the Mediterranean in France at a depth of 476 meters, this dataset contains 442 images with a resolution of 1600×1200 pixels.

4) Lucky Strike Hydrothermal Field Dataset [28]: This dataset features the Eiffel Tower active hydrothermal chimney. It contains 1,061 RGB images.

5) Submarine Fault Scarp and Earthquake Traces Dataset [29]: Located at a depth of 1100 meters, this dataset captures a fault scarp with about 1 meter of co-seismic displacement. Keyframes extracted from video files yielded 590 RGB images.

6) Seafloor Litter Field Dataset [30]: Collected near the French Mediterranean coast at a depth of 600 meters, this dataset consists of 80 RGB images.

As the last four datasets lack corresponding depth maps, they are not suitable for network training. Instead, they are used to test the model, verifying its generalization capabilities.

During the training phase, the model undergoes 200 epochs with an initial learning rate of 0.001, halved at epochs 50, 100, and 150 to avoid local minima. Each iteration uses a batch size of 1, with testing conducted every three epochs. The Adam optimizer is used for optimization, and the depth interval scaling factor is set at 1.06, divided into three stages with hypothesis layers set at 48, 32, and 8 for each stage. Loss weights of 0.5, 1.0, and 2.0 are applied across these stages, respectively, with experimental weights set at $\alpha$=0.5, $\beta$=0.2, $\gamma$=0.2, and $\zeta$=0.1. The computational environment includes PyTorch 2.0.0, Python 3.8 on Ubuntu 20.4, with CUDA 11.8, running on an RTX 3090 GPU with 24GB VRAM and an AMD EPYC 7742 64-Core Processor providing computational support.

## 4.2 Experimental comparison results

To validate the effectiveness of our proposed model, we conducted both quantitative and qualitative experiments on the aforementioned datasets. To enhance the model's stability, it was first pretrained on the BlendedMVS dataset and then fine-tuned using the FLSea dataset. Table 1 presents a summary of the results of our depth estimation method in terms of relative error (Rel), root mean square error (RMSE), and logarithmic error ($\text{Log}_{10}$), compared to PatchmatchNet [31], CasMVSNet [23], and other state-of-the-art depth estimation methods on the FLSea dataset. The results demonstrate that our estimated depth maps consistently outperform those generated by mainstream methods. Compared to the baseline CasMVSNet, our approach achieves a 60.83% reduction in relative error, a 54.87% decrease in RMSE, and a 39.61% reduction in Log10 error, highlighting a substantial improvement in accuracy.

Table 1: Comparison of different methods on the FLSea dataset

| Methods | Rel↓ | RMSE↓ | Log10↓ |
|---|---|---|---|
| PatchmatchNet [31] | 4.2856 | 3.7715 | 0.6990 |
| CasMVSNet [23] | 3.7047 | 3.1136 | 0.5426 |
| GeoMVSNet [17] | 3.4223 | 2.9135 | 0.5317 |
| SeaThru-NeRF [12] | 2.2856 | 2.7715 | 0.4990 |
| Ebner et al. [22] | 2.6975 | 2.8943 | 0.5182 |
| H. Gupta et al. [16] | 1.9835 | 2.3813 | 0.3511 |
| Song et al. [6] | 3.3014 | 2.6510 | 0.3395 |
| Berman et al. [7] | 1.9895 | 2.9731 | 0.3560 |
| UMCS-CS(Ours) | 1.4512 | 1.4053 | 0.3277 |

Figure 4 provides a qualitative comparison of depth maps estimated by our method against those generated by other mainstream methods. The first column displays the input underwater image, columns 2 to 9 present the depth maps generated by other methods, column 10 shows the depth map produced by our method, and column 11 represents the ground truth depth. Columns 2 to 5 indicate that the depth maps estimated by these methods lack overall smoothness and are significantly affected by underwater noise, leading to drastic relative depth changes in non-edge areas. In columns 6 and 7, the estimated depth maps appear excessively smooth, failing to capture finer details due to the limited representation capability of these networks. Columns 8 and 9 reveal ambiguities in depth estimation, particularly with convex and concave surfaces, resulting in inaccurate depth values for the seabed. In contrast, our depth map maintains overall smoothness, demonstrates resilience to underwater noise, and better preserves edge details of underwater objects.

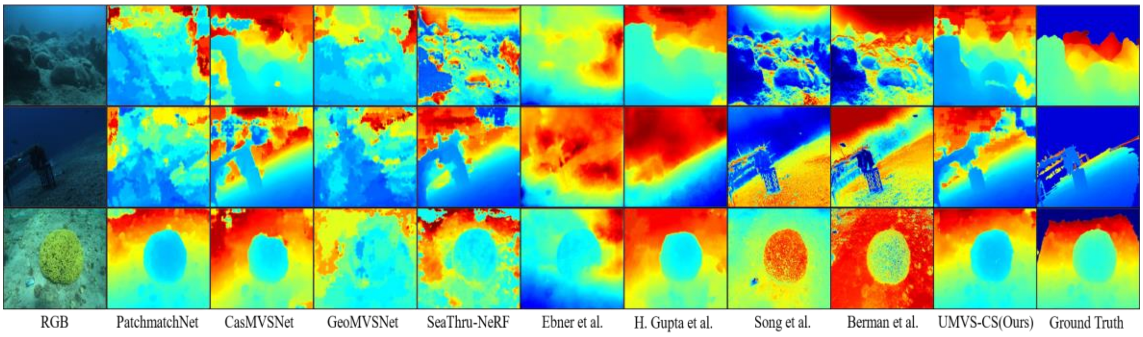

Figure 4: Qualitative comparison of underwater scene depth estimations on the FLSea test set between UMCS-CS and monocular depth estimation models

To further validate the generalization capability of our proposed method across various underwater scenarios, such as those relevant to marine engineering applications (e.g. underwater archaeology, marine resource exploration, underwater ecological monitoring), we tested the pretrained model, trained on the FLSea dataset without further fine-tuning on additional datasets. To evaluate the model's generalization capabilities, we conducted testing experiments on diverse datasets without training, including Torpedo Shipwreck Dataset (476m depth), Eiffel Tower Hydrothermal Vent Dataset (1700m depth), Seafloor Fault and Seismic Trace Dataset (1100m depth), Mediterranean Underwater Litter Dataset (600m depth). These datasets encompass varying depths and distinctly different scenes. Figure 5 shows qualitative comparisons between depth maps predicted by our method and those generated by other models across various scenes.

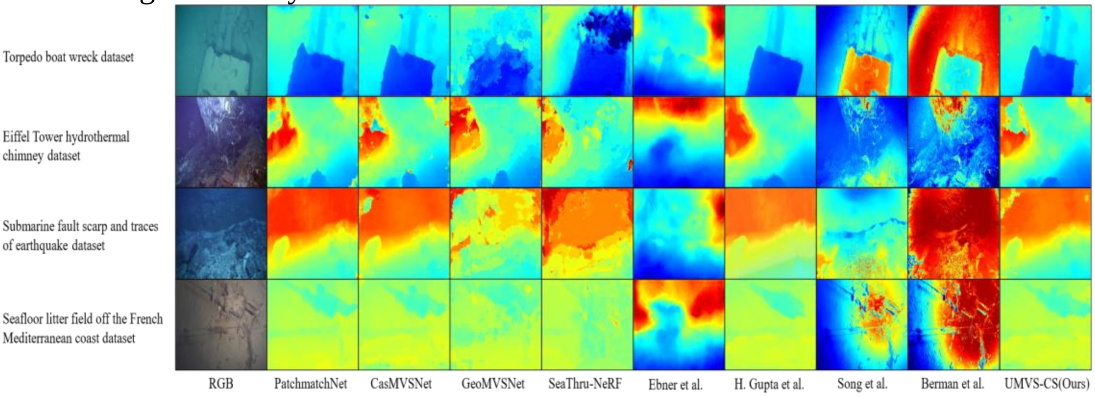
Torpedo boat wreck dataset
Eiffel Tower hydrothermal chimney dataset
Submarine fault scarp and traces of earthquake dataset
Seafloor litter field off the French Mediterranean coast dataset
RGB
PatchmatchNet
CasMVSNet
GeoMVSNet
SeaThru-NeRF
Ebner et al.
H. Gupta et al.
Song et al.
Berman et al.
UMVS-CS(Ours)

Figure 5: Depth maps estimated by the proposed method on the last four datasets, are qualitatively compared with those generated by other monocular depth estimation models

Qualitative comparisons with other models demonstrated that our method predicts relatively accurate depth maps, with minimal depth misalignment and well-defined object edges. This highlights the robustness of our approach in untrained datasets and its ability to generalize effectively across diverse underwater environments. In practical deployment, when reconstructing objects in different marine areas based on captured images, reliable results can be obtained using this model.

## 4.3 Ablation Study

To access the effectiveness of each component of the proposed UMCS-CS model, we conducted a series of quantitative experiments on the FLSea dataset. The quality of the estimated depth maps was evaluated using key metrics, including relative error (Rel), root mean square error (RMSE), and logarithmic error ($\text{Log}_{10}$). Table 2 presents the results of these experiments, demonstrating the impact of each module on the overall performance of the model.

Table 2: Quantitative Comparison of Different Modules on the FLSea Dataset

| Model | Mask | Two-stage feature extraction | Depth range update based on variance | Composite Loss Function | Rel↓ | RMSE↓ | Log10↓ |
|---|---|---|---|---|---|---|---|
| Backbone | | | | | 3.7047 | 3.1136 | 0.5426 |
| Ours-A | √ | | | | 3.3674 | 2.9428 | 0.5074 |
| Ours-B | √ | √ | | | 2.4149 | 1.9132 | 0.4541 |
| Ours-C | √ | √ | √ | | 2.3674 | 1.8428 | 0.4487 |
| Ours | √ | √ | √ | √ | 1.4512 | 1.4053 | 0.3277 |

Our baseline model was progressively enhanced by introducing new components: mask filtering, a two-stage feature extraction process, variance-based depth range updating, and a composite loss function. As shown in Table 2, each addition contributed to improved prediction accuracy. For example, the incorporation of the two-stage feature extraction module significantly reduced the relative error and RMSE, while the variance-based depth range updating further refined the accuracy of depth predictions. The composite loss function effectively enhanced the model’s generalization performance across various test scenarios, further confirming the contribution of each module to the overall performance improvement.

The ablation study shows that each component plays a critical role in improving the model’s depth estimation performance. The final model (Ours) incorporates all modules, achieving the lowest relative error, RMSE, and $Log_{10}$ values, demonstrating its superior accuracy and robustness in underwater monocular depth estimation tasks.

## 5. Conclusions

In this paper, we propose an underwater monocular continuous stereoscopic network (UMCS-CS) based on a cascade structure to enhance depth estimation in underwater environments. The network improves feature representation by combining original optical images with local and global features, which is conducive to distinguishing object edges. By introducing a mask to remove the spot area and adjusting the variance of the current depth estimation, the depth estimation range is appropriately expanded, taking into account the isolated, discontinuous and irregular "spots". A cascade structure is designed to extract multi-layer change information from optical images, and a composite loss function is introduced to improve the generalization ability of the model.

Experimental results demonstrate that the UMCS-CS method significantly improves depth map estimation accuracy for various underwater terrains, structures, and environments. The model achieves a notable reduction in relative error, root mean square error, and logarithmic error compared to existing state-of-the-art methods. Additionally, it exhibits strong generalization performance across multiple datasets, including underwater archaeological sites, marine resource exploration areas, and ecological monitoring scenarios.

In summary, the UMCS-CS network effectively performs depth estimation in marine engineering applications, such as detecting underwater for pipelines laying or determining distances to underwater archaeological structures. Despite these advances, there remains room for further improvement. Future work will explore enhancements to the cost structure and regularization components of the proposed method, aiming to reduce the model's reliance on high memory usage.

## Acknowledgements

This research is supported by the National Nature Science Foundation of China (Grant No. 62301302).

## References

*[1] Zhang, S., Zhao, S., An, D., Liu, J., Wang, H., Feng, Y., Li, D., Zhao, R. Visual SLAM for underwater vehicles: A survey[J]. Comput. Sci. Rev., 2022, 46:100510.*

*[2] Hu, Q., Zhu, H., Yu, M., Fan, Z., Zhang, W., Liu, X., Li, Z. A novel 3D detection system with target keypoint estimation for underwater pipelines[J]. Ocean Engineering, 2024, 309:118319.*

*[3] Zhang, X., Bian, X., Yan, Z. Underwater Docking of AUV with the Dock and Virtual Simulation[C]. Advanced Materials Research, 2010, 159:371-376.*

*[4] Sahu, P., Gupta, N., Sharma, N. A Survey on Underwater Image Enhancement Techniques[J]. International Journal of Computer Applications, 2014, 87:19-23.*

*[5] Bello, O., Zeadally, S. Internet of underwater things communication: Architecture, technologies, research challenges and future opportunities[J]. Ad Hoc Networks, 2022, 135:102933.*

*[6] Song, W., Wang, Y., Huang, D., Tjondronegoro, D. A Rapid Scene Depth Estimation Model Based on Underwater Light Attenuation Prior for Underwater Image Restoration[C]. PCM 2018, 2018, 678–688.*

*[7] Berman, D., Levy, D., Avidan, S., Treibitz, T. Underwater Single Image Color Restoration Using Haze-Lines and a New Quantitative Dataset[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018, 43:2822-2837.*

*[8] Chen, Y., Li, Q., Gong, S., Liu, J., Guan, W. UV3D: Underwater Video Stream 3D Reconstruction Based on Efficient Global SFM[J]. Applied Sciences, 2022, 5918.*

*[9] Schönberger, J.L., Zheng, E., Frahm, J.-M., Pollefeys, M. Pixelwise View Selection for Unstructured Multi-View Stereo[C]. ECCV 2016, 2016, 501–518.*

*[10] Romanoni, A., Matteucci, M. TAPA-MVS: Textureless-Aware PAtchMatch Multi-View Stereo[C]. IEEE/CVF ICCV 2019, 2019, 10412-10421.*

*[11] Zhu, Z., Li, X., Wang, Z., He, L., He, B., Xia, S. Development and Research of a Multi-Medium Motion Capture System for Underwater Intelligent Agents[J]. Applied Sciences, 2020, 10:6237.*

*[12] Levy, D., Peleg, A., Pearl, N., Rosenbaum, D., Akkaynak, D., Korman, S., Treibitz, T. SeaThru-NeRF: Neural Radiance Fields in Scattering Media[C]. IEEE/CVF CVPR, 2023, 56–65.*

*[13] Ye, X., Zhang, J., Yuan, Y., Xu, R., Wang, Z., Li, H. Underwater Depth Estimation via Stereo Adaptation Networks[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2023, 33:5089–5101.*

*[14] Li, K., Wang, X., Liu, W., Qi, Q., Hou, G., Zhang, Z., Sun, K. Learning Scribbles for Dense Depth: Weakly Supervised Single Underwater Image Depth Estimation Boosted by Multitask Learning[J]. IEEE Transactions on Geoscience and Remote Sensing, 2024, 62:1–15.*

*[15] Gupta, H., Mitra, K. Unsupervised Single Image Underwater Depth Estimation[C]. IEEE ICIP, 2019, 624–628.*

*[16] Nagamatsu, G., Takamatsu, J., Iwaguchi, T., Thomas, D.G., Kawasaki, H. Self-calibrated dense 3D sensor using multiple cross line-lasers based on light sectioning method and visual odometry[C]. IEEE/RSJ IROS, 2021, 94–100.*

*[17] Zhang, Z., Peng, R., Hu, Y., Wang, R. GeoMVSNet: Learning Multi-View Stereo with Geometry Perception[J]. IEEE/CVF CVPR, 2023, 21508–21518.*

*[18] Qi, X., Liu, Z., Liao, R., Torr, P.H.S., Urtasun, R., Jia, J. GeoNet++: Iterative Geometric Neural Network with Edge-Aware Refinement for Joint Depth and Surface Normal Estimation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 969–984.*

*[19] Marques, F.M., Castro, F., Parente, M., Costa, P. A Hybrid Framework for Uncertainty-Aware Depth Prediction in the Underwater Environment[J]. IEEE ICARSC, 2020, 102–107.*

*[20] Li, Q., Wang, H., Xiao, Y., Yang, H., Chi, Z., Dai, D. Underwater Unsupervised Stereo Matching Method Based on Semantic Attention[J]. J. Mar. Sci. Eng., 2024, 12:1123.*

*[21] Ebner, L., Billings, G., Williams, S. Metrically Scaled Monocular Depth Estimation through Sparse Priors for Underwater Robots[C]. ICRA 2024, Yokohama, Japan.*

*[22] Ye, X., Li, Z., Sun, B., Wang, Z., Xu, R., Li, H., Fan, X. Deep Joint Depth Estimation and Color Correction From Monocular Underwater Images Based on Unsupervised Adaptation Networks[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2020, 3995–4008.*

*[23] Gu, X., Fan, Z., Zhu, S., Dai, Z., Tan, F., Tan, P. Cascade Cost Volume for High-Resolution Multi-View Stereo and Stereo Matching[C]. IEEE/CVF CVPR, 2019, 2492–2501.*

*[24] Sturm, P. Pinhole Camera Model[M]. In Computer Vision: A Reference Guide. Springer International Publishing, 2021, 983-986.*

*[25] Yao, Y., Luo, Z., Li, S., Zhang, J., Ren, Y., Zhou, L., Fang, T., Quan, L. BlendedMVS: A Large-Scale Dataset for Generalized Multi-View Stereo Networks[C]. IEEE/CVF CVPR, 2019, 1787–1796.*

*[26] Randall, Y., Treibitz, T. FLSea: Underwater Visual-Inertial and Stereo-Vision Forward-Looking Datasets[J]. The International Journal of Robotics Research, 2023.*

*[27] Arnaubec, A., Raugel, E. Torpedo boat wreck (Mediterranean, 43.124N;6.523E): Imagery and 3D model[J]. SEANOE, 2021.*

*[28] Matabos, M., Arnaubec, A. Eiffel Tower hydrothermal chimney (Lucky Srike Hydrothermal Field, Mid Atlantic Ridge): 3D scene and imagery[J]. SEANOE, 2015.*

*[29] Arnaubec, A., Escartin, J. Submarine fault scarp and traces of earthquake (Roseau Fault, French Antilles): 3D scene and imagery[J]. SEANOE, 2017.*

*[30] Arnaubec, A., Raugel, E. Seafloor litter field off the French Mediterranean coast (43.078°N;6.458°E): 3D scene and imagery[J]. SEANOE, 2017.*

*[31] Wang, F., Galliani, S., Vogel, C., Speciale, P., Pollefeys, M. PatchmatchNet: Learned Multi-View Patchmatch Stereo[C]. IEEE/CVF CVPR, 2020, 14189–14198.*