

UW Deep SLAM-CNN Assisted Underwater SLAM

Chinthaka Amarasinghe^{1*}, Asanga Ratnaweera², Sanjeeva Maitripala³

¹*Department of Science & Technology, Uva Wellassa University, Sri Lanka*

^{2,3}*Department of Mechanical Engineering, University of Peradeniya, Sri Lanka*

Abstract – Underwater simultaneous localization and mapping (SLAM) poses significant challenges for modern visual SLAM systems. The integration of deep learning networks within computer vision offers promising potential for addressing these difficulties. Our research draws inspiration from deep learning approaches applied to interest point detection and matching, single image depth prediction and underwater image enhancement. In response, we propose 3D-Net, a deep learning-assisted network designed to tackle these three tasks simultaneously. The network consists of three branches, each serving a distinct purpose: interest point detection, descriptor generation, and depth prediction. The interest point detector and descriptor generator can effectively serve as a front end for a classical SLAM system. The predicted depth information is akin to a virtual depth camera, opening up possibilities for various applications. We provide quantitative and qualitative evaluations to illustrate some of these potential uses. The network was trained in several steps, using in-air datasets and followed by generated underwater datasets. Further, the network is integrated into feature-based SALM systems ORBSLAM2 and ORBSSLAM3, providing a comprehensive assessment of its effectiveness for underwater navigation.

Keywords – Autonomous underwater vehicle, deep-SLAM, machine learning, underwater navigation.

I. INTRODUCTION

Autonomous driving, robot navigation, trajectory tracking, and path planning are among the pivotal tasks that heavily depend on a robot's ability to accurately determine its position within an unfamiliar environment. Addressing this challenge effectively is achieved through simultaneous localization and mapping (SLAM). SLAM systems play a vital role in enabling localization and mapping capabilities in unexplored territories, thus becoming an indispensable requirement for mobile and autonomous robots. Since it was initially introduced in 1986, SLAM received a lot of attention from researchers and rapidly developed in a wide range of fields, including robotics and virtual reality [1]. Contemporary visual SLAM frameworks comprise four key modules [2]: the frontend for visual odometry (VO), backend optimization, mapping, and loop closure detection. While conventional SLAM systems have progressed rapidly and demonstrated exceptional performance under favourable conditions, numerous systems encounter challenges when operating in dynamic and outdoor environments. Consequently, the underwater environment poses a significant challenge for modern visual SLAM systems,

prompting efforts to overcome these obstacles. The integration of deep learning in computer vision presents promising opportunities for addressing such difficulties effectively. By employing deep learned networks, it becomes feasible to substantially mitigate the issues that conventional SLAM systems struggle to overcome.

Machine learning has showcased remarkable achievements in various computer vision tasks, encompassing 3D reconstruction [3], image recognition [4], semantic comprehension [5], and image matching [6]. Moreover, numerous approaches leveraging deep learning have shown promising outcomes in tackling SLAM and VO challenges. However, despite the emergence of end-to-end learning methodologies, they still lag behind conventional systems in terms of accuracy and practicality [7], [8].

In addition to end-to-end learning, several researchers [9], [10] propose employing deep learning as an assistive system by replacing certain modules within conventional VSLAM. With this approach in mind, our focus was on replacing the underperforming modules, namely interest point detection, descriptor generation, and depth generation, in underwater Visual SLAM (VSLAM) systems. We draw inspiration from previous deep learning research focusing on interest point detection and matching, single image depth predictions, and underwater image enhancement. While interest point detection and tracking are considered low-level tasks, others are regarded as high-level tasks, with each system demonstrating outstanding performance as standalone networks. In contrast to employing separate networks for each task, our approach involves designing a unified network with multiple branch outputs, addressing image enhancement, interest point detection, descriptor generation, and depth prediction. This integrated network, referred to as 3D-Net, comprises three branches that produce outputs for interest point detection, descriptor generation, and depth prediction. The predicted depth serves as a virtual depth camera, opening up various potential applications.

Figure 1 illustrates the construction of a dense map for underwater trajectory by projecting the predicted depth images based on estimated camera poses. Additionally, we explored other intriguing applications, such as metric scale prediction for monocular SLAM and virtual RGBD-SLAM.

* Corresponding author's e-mail: chinthakaa@uwu.ac.lk

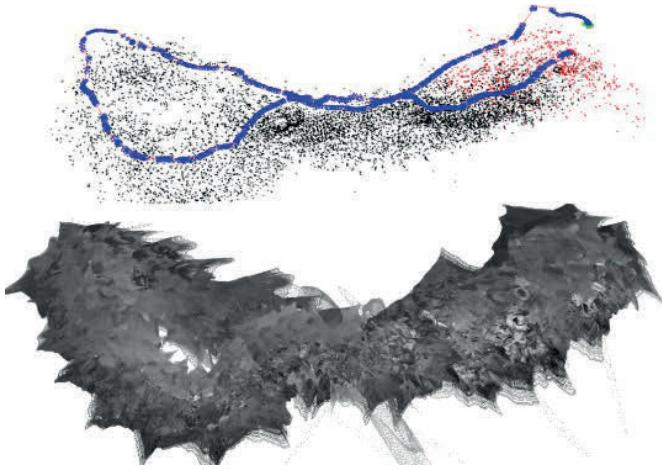


Fig. 1. Sparse map and the Dense map generated for Aqualoc harbour 01 Sequence with modified 3D-Net+ ORBSLAM3 in monocular inertial SLAM.

The 3D-Net takes greyscale images and sparse depth inputs as its inputs. Greyscale images were chosen over RGB due to their superior performance in the initial testing of the detector and descriptor branches. Sparse depth inputs are obtained from the mapping module of a feature-based VSLAM system in real-time. The training of the proposed network is conducted in two stages. Initially, it is trained using datasets captured in an in-air environment. Subsequently, the network is further fine-tuned using generated underwater datasets to adapt it to underwater conditions. Performance evaluation of the proposed 3D-Net also followed the same approach using air data and real underwater data. Moreover, the integrated network is seamlessly incorporated into feature-based SLAM systems, namely ORBSLAM2[11] and ORBSLAM3[12]. The trajectory tracking performance of these systems is evaluated and compared against benchmark approaches. The key contributions of this paper can be summarised as follows:

1. Introduction of the 3D-Net, a Visual Odometry (VO)-assisted network designed to generate outputs for interest point detection, descriptor generation and dense depth mapping.
2. Utilization of a generative adversarial network (GAN) to generate synthetic underwater-like images, which are then used to train the 3D-Net and improve its performance in underwater environments.
3. Seamless integration of the 3D-Net into the ORBSLAM2 and ORBSLAM3 feature-based SLAM systems.
4. Comparative analysis of the proposed 3D-Net with state-of-the-art approaches, providing insights into its performance and advancements in the field.

The subsequent sections of the paper are organised as follows. Section 2 provides a concise overview of the present state of deep learned vision navigation. In Section 3, the 3D-Net, our proposed method, is elaborated upon in detail. Section 4 outlines the experimental setup and the training procedure employed. Furthermore, Section 5 presents the experimental results, along with a comparative evaluation against state-of-the-art approaches. Lastly, Section 6 concludes the paper by discussing future directions.

II. RELATED WORK

In this review, our investigation focuses on three paradigms: deep-learn assisted visual odometry, interest point detection and tracking, and single image depth prediction. For each paradigm, we discuss the relevant work that has been conducted in these areas.

A. SLAM with Depth Prediction

Extensive evidence has demonstrated the significant advantages of Convolutional Neural Networks (CNN) in single image depth prediction [13]. In their study, the authors utilized the predicted depth values to generate a dense reconstruction of the environment. Their approach involved training a network to produce depth predictions by using RGB images as inputs and incorporating sparse depth obtained from a feature-based SLAM system in real time. Additionally, the authors addressed the challenge of scale ambiguity in monocular SLAM by leveraging the predicted depth images. In the domain of LSD-SLAM, [14] introduced a real-time SLAM system called CNN-SLAM in 2017. By training a depth estimation network model, they achieved accurate depth mapping. This depth map was subsequently integrated into the pose estimation tasks within the SLAM system. Further refinement of depth values took place within the optimization module of the SLAM system. Notably, the framework also incorporated a CNN-based semantic segmentation module, enhancing the VSLAM system capacity to perceive complex visual information. Similar research endeavours utilizing neural networks for depth estimation include Code-SLAM [15] and DVSO [16], both of which utilize stereo cameras to estimate depth information.

B. SLAM with Learned Feature Detection and Matching

Convolutional models, in particular, are a subset of deep learning techniques that have given rise to novel key point detection and description techniques that promise to outperform traditional algorithms in benchmark tests. Surveys on deep learned detector and descriptor methods can be found in [17], [18]. Convolutional models learn about the features based on supervision rather than manually creating them; hence, their performance is highly dependent on ground truth data.

Traditional appearance-based descriptors, which are frequently employed for local mapping, provide locally consistent maps with high performance, as demonstrated by ORBSLAM2 [11], SVO [19], and DSO [20]. However, systems that rely on visual keypoint descriptors, such as SIFT [21], SURF [22], BRIEF [23], and ORB [24], frequently fail to relocalize due to turbidity, scattering, poor illumination, and short occlusion in an underwater environment. There are few notable deep-learning interest point detectors and descriptors [25]–[28], but none of them addresses the difficulties that arise underwater specifically. SuperPoint [25] is a self-supervised approach that shows interesting results on benchmark datasets. It adopts a new strategy by employing homographies of images to learn correspondence while self-labeling a large collection of images. Despite being remarkably effective for its simplicity, this method is constrained by its reliance on static images, which prevents it from learning from actual changes in lighting

and correspondence in challenging nonplanar settings. This approach uses a straightforward homography model as its backend, which generates dynamically. Superpoint is later integrated into a SLAM system [29] and SuperpointVO [30] that demonstrated comparative results compared to traditional SLAM systems.

A self-improving deep-learning VO was proposed in [31] which additionally predicts a keypoint stability score compared to Superpoint. The system is composed of learned based front end and conventional backend. The suggested approach can recognise optimal points for VO and inhibit unstable points, such as those brought on by moving objects. In order to achieve feature extraction in a conventional backend based on ORB-SLAM [28] architecture, LIFT-SLAM [3] proposes a unique hybrid VSLAM technique based on the LIFT [22] network. Authors asserted that, compared to a conventional VSLAM algorithm, LIFT-SLAM is more reliable without sacrificing precision.

C. SLAM with Multiple Networks

Another interesting line of work is the use of multiple networks to obtain different outcomes in a multi-net multi-branch network. Most of the above-discussed networks are single networks, while some of them contain multi-branch outputs. Superpoint [25] has two branches for detection and descriptor generation, whereas CNN-SLAM [14] has two branches for depth prediction and semantic labelling. However, all of these tasks involve either low-end tasks or high-end tasks. In addition to those studies, D3VO [32] uses two separate networks which are used each for low-level and high-level tasks. This approach makes use of deep learning in three different ways: depth estimation, pose estimation, and uncertainty estimation. To enhance both front-end tracking and

back-end nonlinear optimization efficiency, the prediction depth, posture, and uncertainty are tightly merged into a direct VO. Another system which leverages deep learning and geometry methods is shown in DF-VO [33]. As intermediate outputs, the system forecasts optical flow and single-view depth using deep neural networks to create 2D-2D/3D-2D correspondences for camera pose estimation. Another work presented in [34] used multi-networks to unsupervisory address single view depth prediction, camera motion estimation, optical flow, and segmentation, which are four interrelated low-level vision issues.

III. DEEP ASSISTED NETWORK

3D-Net is a fully-convolutional neural network architecture, which outputs several desired predictions in a single pass. The network operates on a full-sized image and produces feature point detections, fixed length descriptors, and a predicted depth map. The proposed 3D-Net is shown in Fig. 2, and further information is given in Table I. We are inspired by several performing networks for each branch [5]–[9]. 3D-Net follows encoder-decoder architecture. The model has two sets of encoder parts. The first part is a shared encoder that processes and reduces the dimensionality of the input image by a factor of eight. After the first set of encoding, the architecture splits into three branches. Two branches were separated out and continued to be used as decoders for feature point detection and feature point description. The third branch was further encoded, which is the second part of the encoding, and reduced the dimensionality of the original image size by a factor of 32. This branch is then decoded back to deliver the depth map of the scene.

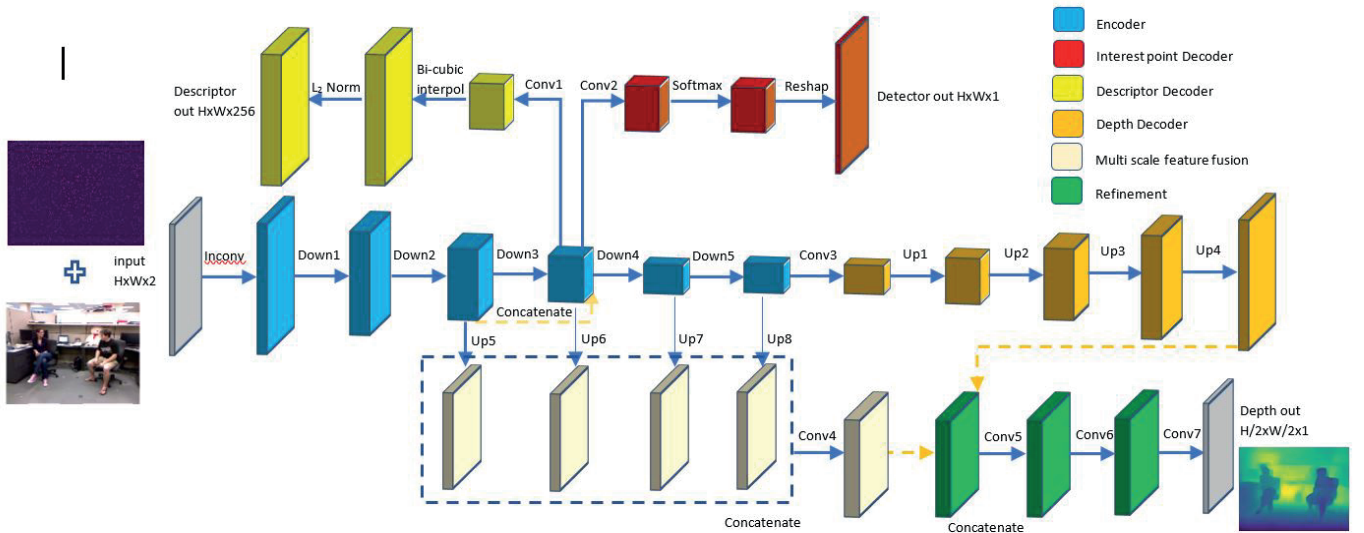


Fig. 2. Diagram of the proposed 3D-Net architecture.

TABLE I
DETAILS OF THE 3D-NET

Layer	Input size	Output size	In/C	Out/C
Incov	$H \times W$	$H \times W$	2	64
Down1	$H \times W$	$H/2 \times W/2$	64	128
Down2	$H/2 \times W/2$	$H/4 \times W/4$	128	256
Down3	$H/4 \times W/4$	$H/8 \times W/8$	256	256
Down4	$H/8 \times W/8$	$H/16 \times W/16$	512	1020
Down5	$H/16 \times W/16$	$H/32 \times W/32$	1020	2040
Conv1	$H/8 \times W/8$	$H/8 \times W/8$	256	256
Bi-cubic Interpolate	$H/8 \times W/8$	$H \times W$	256	256
L2 Norm	$H \times W$	$H \times W$	256	256
Conv2	$H/8 \times W/8$	$H/8 \times W/8$	256	65
Softmax	$H/8 \times W/8$	$H/8 \times W/8$	65	65
Reshape	$H/8 \times W/8$	$H \times W$	64	1
Conv3	$H/32 \times W/32$	$H/32 \times W/32$	2040	1020
Up1	$H/32 \times W/32$	$H/16 \times W/16$	1020	512
Up2	$H/16 \times W/16$	$H/8 \times W/8$	512	256
Up3	$H/8 \times W/8$	$H/4 \times W/4$	256	128
Up4	$H/4 \times W/4$	$H/2 \times W/2$	128	64
Up5	$H/4 \times W/4$	$H/2 \times W/2$	256	16
Up6	$H/8 \times W/8$	$H/2 \times W/2$	512	16
Up7	$H/16 \times W/16$	$H/2 \times W/2$	1020	16
Up8	$H/32 \times W/32$	$H/2 \times W/2$	2040	16
Conv4	$H/2 \times W/2$	$H/2 \times W/2$	64	64
Conv5	$H/2 \times W/2$	$H/2 \times W/2$	128	128
Conv6	$H/2 \times W/2$	$H/2 \times W/2$	128	128
Conv7	$H/2 \times W/2$	$H/2 \times W/2$	128	128

A. Encoder Block

The 3D-Net architecture uses a set of residual network blocks [35], which we call the ‘Resnet block’ as encoder to reduce the dimensionality of the image. We used the residual network architecture [35] as it has proven performance in image classification and further usage for underwater image enhancement tasks. Each Resnet block consists of convolutional layers, skipped connections, spatial down sampling via pooling, and nonlinear activation functions. Figure 3 shows a block diagram of the Resnet block and the ‘down’ block used in the encoder. The first part of the encoder uses three ‘down’ blocks, down sampling to $H_1 = H/8$ and $W_1 = W/8$, and a depth of 256 channels for an image sized $H \times W$. The second part of the encoder uses two ‘down’ blocks, down sampling to $H_2 = H/32$ and $W_2 = W/32$, and a depth of 2040 channels for an image size $H \times W$. The ‘incov’ block contains one Resnet block without a max-pooling layer.

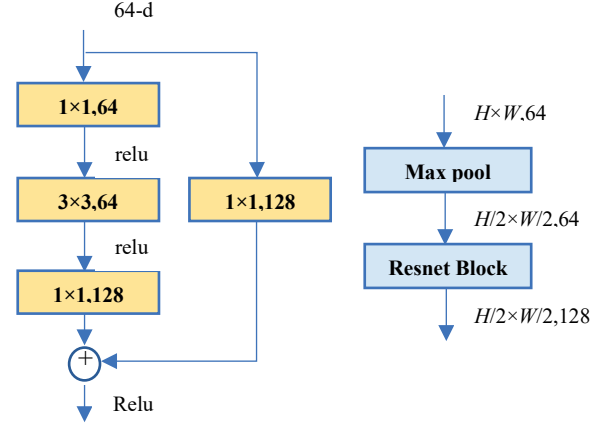


Fig. 3. Left image: Diagram of a Resnet Block with 64 input and 128 output channels. Right Image: Diagram of a ‘down1’ module which down sampling an image by a factor of 2.

B. Deep Detector Head and Deep Descriptor Head

We are inspired by the work of Superpoint [25] and have adapted a similar architecture to the 3D-Net detection and descriptor branches. However, instead of using VGG [36] style network in the original Superpoint, we employed Residual network [35] architecture for the construction of the main encoder, as it is becoming a common choice in underwater image enhancement networks [5]. The detector decoder computes $H_1 \times W_1 \times 256$ to $H_1 \times W_1 \times 65$. The last channel was considered a “no interest point” and fed to the dustbin. After a channel-wise softmax $H_1 \times W_1 \times 64 \Rightarrow H \times W$ reshape is performed. Similarly, the descriptor decoder computes $H_1 \times W_1 \times 256$ to $H \times W \times 256$ by performing bi-cubic interpolation of the descriptor and then L2-normalizes the activations to be unit length. Conv1 and Conv2 consist of 3×3 convolutional layer followed by 1×1 convolutional layer. Readers interested in further exploration can refer to the original research paper on Superpoint for comprehensive insights.

C. Depth Prediction Head

Our implementation utilizes a depth prediction network, as described in [37] comprising four modules: an encoder (E), a decoder (D), a multiscale feature fusion module (MFF), and a refinement module (R). Similar to other depth prediction networks employing skipped connections, the work in [37] incorporates a multiscale fusion module and a refinement module. Notably, this approach demonstrates superior performance, particularly in scenarios involving small objects and object boundaries. Utilizing the existing network, we substitute the initial encoder with our custom design. Our modified feature extraction framework (MFF) operates at distinct scales, precisely $1/4$, $1/8$, $1/16$, and $1/32$. To accomplish the upscaling of the ultimate feature from the encoder while simultaneously reducing the channel count, the decoder employs four up-projection modules [38]. The up-sampling process is performed on a channel-wise basis. The up-sampled results are then concatenated and undergo further transformation through a convolutional layer (conv4). This operation yields the output of the MFF module, resulting in an

output of $H/2 \times W/2 \times 64$. Conv4 consists of a 5×5 convolutional layer followed by batch normalization. Fused multiscale features from the MFF and decoded features are then concatenated in their channels and feed to the refinement module, which contains three convolutional layers (conv5, conv6, and conv7) to provide the final prediction. Conv5 and conv6 both consist of a 5×5 convolutional layer followed by batch normalization, and conv7 consists of only one 5×5 convolutional layer.

D. Sparse-Depth Input

In contrast to direct-based SLAM systems, feature-based SLAM systems produce sparse maps. The challenge of generating a dense map using the output of a feature-based SLAM system has been an intriguing and difficult topic. We attempt to tackle this problem by leveraging the output from the third branch of our network. However, the precision and reliability of these predictions are currently insufficient for practical use [39].

The researchers in [39] explore the use of sparse depth measurements coupled with RGB data to reconstruct depth in full resolution to solve the basic constraints of RGB-based depth estimation. They have shown 50 % increase in accuracy of the predicted depth map. As the SLAM system progresses, sparse depth measurements are estimated from successive feature point matches and further optimized in local bundle adjustment. Therefore, we can potentially utilize the optimized sparse map as an input to the network. Building upon the approach presented in [39], we incorporated sparse depth information into the 3D-Net by concatenating it with the greyscale images. This concatenation resulted in an input tensor of dimensions $H \times W \times 2$, combining both greyscale and sparse depth data.

E. Loss Functions

Given that the proposed network generates three branches, we utilize three distinct loss functions. These loss functions are employed for the interest point detector $\mathcal{L}_p$ [25], the descriptor $\mathcal{L}_d$ [25] and the depth map predictor $\mathcal{L}_m$ [39]. Although the specific loss functions are derived from [25] and [39], for clarity, we provide their illustration below.

Detector Loss

$$\mathcal{L}_p = \mathcal{L}(X, Y) + \mathcal{L}(X', Y'), \quad (1)$$

(X', Y') represent synthetically warped image from (X, Y) which have both (X, Y) and (X', Y') and contain pseudo-ground truth interest point locations, $y_{hw} \in Y$ are the ground truth correspondence from homography H which relates the two images. H is generated randomly on the fly. The interest point detector loss function $\mathcal{L}_p$ is a fully convolutional cross-entropy loss over the cells $x_{hw} \in X$. The loss is

$$\mathcal{L}_p(X, Y) = \frac{1}{H_c H_w} \sum_{h=1}^{H_c W_c} l_p(x_{hw}; y_{hw}), \quad (2)$$

where

$$l_p(x_{hw}; y_{hw}) = -\log\left(\frac{\exp(x_{hwy})}{\sum_{k=1}^{65} \exp(x_{hwk})}\right).$$

Descriptor Loss

All pairs of descriptor cells from the first image $\mathbf{d}_{hw} \in D$ and the second image $\mathbf{d}'_{h'w'} \in D'$ are subjected to the descriptor loss. It is possible to write the descriptor loss, correlation between the cell h, w and the h', w' cell as follows:

$$\mathcal{L}_d = \frac{1}{(H_c W_c)} \sum_{h=1}^{H_c W_c} \sum_{w=1}^{H_c W_c} l_d(d_{hw}, d'_{h'w'}; s_{hwh'w'}), \quad (3)$$

where

$$l_d(d, d'; s) = \lambda_d * s * \max(0, m_p - d^T d') + (1 - s) * \max(0, d^T d' - m_n),$$

λ_d is the weighting term, m_p and m_n are positive and negative margins respectively. s denotes the correspondence for a pair of images.

Depth Loss

Depth loss was taken as the $\mathcal{L}_1$ loss, as it has proven for its simplicity and performance [39].

$$\mathcal{L}_m = \frac{1}{n} \sum_{i=1}^n |e_i|. \quad (4)$$

Final loss is computed as the weighted summation of the above three losses where

$$\mathcal{L}_{\text{Total}} = \mathcal{L}_p + \alpha \mathcal{L}_d + \beta \mathcal{L}_m. \quad (5)$$

IV. EXPERIMENTAL SETUP

A. Training with Air Images

Training was carried out sectionally, where detector and descriptor branches were trained as the first step. Training was conducted in a self-supervised fashion following steps from [25]. The script presented in [40] was modified and used to train the detector branches by generating ground truth data from the original Superpoint network [25]. COCO [41] and NYU Depth V2 [42] datasets were used to train descriptor and detector branches jointly with homographic adaptation [25]. In the second step, the depth prediction branch was trained by loading the pre-trained values from the first step and further loading the pre-trained values from [37], which were loaded into the D, MFF, and R modules. We keep the detector, descriptor, and first part of the encoder as freeze layers in this training session. The training was conducted with NYU depth V2 [42] dataset in a supervised manner. As the third and final step, all three branches were trained jointly by utilizing pre-trained values in steps one and two using NYU depth V2 [42] dataset. Ground truth for detector and descriptor were generated from the pertained network obtained in the first step.

B. Training on Synthetic Underwater Images

Training with underwater scenarios requires underwater images, their respective ground truth interest point detection, and the respective underwater ground truth depth map. This combination of data is hard to come by and almost not possible under current technology, as the depth camera has a limited range and noisy measurements in underwater [17], [18]. Therefore, we develop a synthetic approach to generate underwater-like images from air images with the help of a GAN. This approach was inspired by the work in [45], [46] and the technique has been proven to demonstrate successive results with underwater image enhancements [46], [5], [47]. We train the adversarial network UWGAN presented in [45] and generate underwater-like images. The generated underwater images together with the corresponding air images were then used to train the 3D-Net by following three steps in Section IV-A. Figure 4 shows the training dataset with generated underwater-like images generated from the UWGAN [45] network.

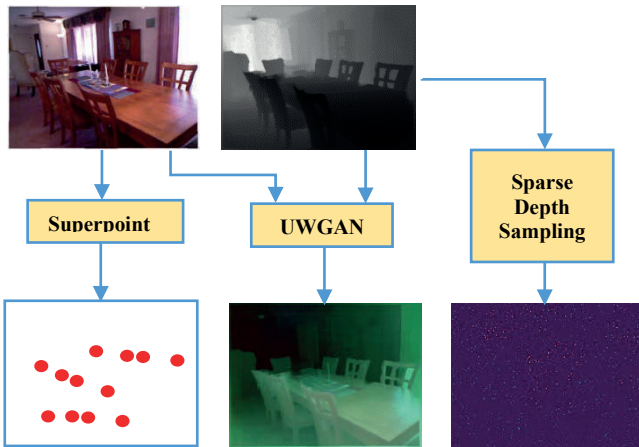


Fig. 4. Underwater-like image generation with ground truth data: a – Generated ground truth interest points, b – Underwater like images, c – Sampled Sparse depth inputs.

C. Depth Sampling

We incorporate the depth sampling strategy used in [39], but instead of random sampling, FAST [48] corner detectors were used to select interest points and create the input sparse depth D from the ground truth depth image D^* on the fly. This injects different number of points and simulates the exact scenario as in VO.

D. Data Augmentation

The following data augmentation techniques were used and separately applied online to each sample (an RGB picture and the associated depth map):

- *Colour jitter*: brightness, contrast, and saturation values of the rgb image are randomly scaled by $c \in [0.6, 1.4]$;
- *Scale*: the rgb images are scaled by a random number $s \in [1, 1.5]$, and depths are divided by s ;
- *Flip*: the rgb image and the depth image are both horizontally flipped;
- *Rotation*: the rgb image and the depth image are both rotated by a random degree $r \in [-5, 5]$;
- *Color normalization*: rgb is normalized through mean subtraction and division by standard deviation and then converted to greyscale images.

V. EXPERIMENTAL RESULT

Training sessions were carried out initially with air datasets and followed by generated underwater-like datasets. We conducted experiments sectionally for detectors, descriptors, and depth map generation. The results were taken at two stages according to the training scenarios: air and underwater.

A. Evaluation of Detector & Descriptor

The first set of results were obtained after training with air images. HPatches [49] dataset was used to evaluate the detector and descriptor performance with respect to the original Superpoint and also compared to classical detector descriptor combinations, such as ORB [24] and SIFT [21]. The original Superpoint network was also trained from the beginning with a similar setup. HPatches [49] contained different illuminations and different scene images. The second set of results was obtained after training with underwater-like images generated with UWGAN [45] network. For evaluation, we used two underwater datasets: a real underwater dataset with different water types containing several levels of turbidity [50] (we call this as UW-TURBID) and [51] a real underwater dataset with synthetically generated ground truth containing a large set of images (we called this UW-BENCHMARK). The generated results are tabulated in Tables II and III. Figure 5 presents the qualitative results obtained from the UW-TURBIT dataset. The images showcase the successive detection and matching of interest points using various detector and descriptor combinations, in comparison with the results achieved by the 3D-Net approach (one yellow colored line shows a successive match between clear image and the turbid image). For air images, the proposed 3D-Net achieved similar performance as the Superpoint network but outperformed in underwater datasets. It is only the 3D-Net that is able to track all 72 images in UW-TURBIT. In both cases, 3D-Net trained with underwater-like images outperforms Superpoint and classical detector and descriptor combination by a large margin in underwater scenarios.

TABLE II
ESTIMATION ON HPATCHES, SP – SUPERPOINT.
RESULT OBTAINED AFTER TRAINING WITH COCO DATASET

	Homography Estimation			Detector Metrics		Descriptor Matrices	
	$\epsilon = 1$	$\epsilon = 3$	$\epsilon = 5$	repeatability	MLE	mAP	M.score
SP	0.479	0.755	0.8	0.571	1.27	0.685	0.315
SIFT	0.663	0.791	0.829	0.512	1.16	0.739	0.261
FAST+ORB	0.329	0.543	0.608	0.591	1.05	0.515	0.155
3D-Net	0.472	0.755	0.813	0.569	1.27	0.681	0.314

TABLE III
ESTIMATION ON UW-TURBIT. SP – SUPERPOINT TRAINED WITH COCO,
SP-UW – SUPERPOINT TRAINED WITH COCO AND UNDERWATER DATASETS,
3D-NET AND 3D-NET-UW FOLLOWS THE SAME

	Homography Estimation			Detector Metrics		Descriptor Matrices		# images
	$\epsilon = 1$	$\epsilon = 3$	$\epsilon = 5$	Repeatability	MLE	mAP	M.Score	
SP	0.85	0.925	0.925	0.753	0.614	0.885	0.725	70
SP-UW	0.85	0.925	0.925	0.762	0.604	0.852	0.726	73
SIFT	0.819	0.847	0.861	0.617	0.698	0.892	0.597	72
FAST+ORB	0.803	0.91	0.91	0.707	0.446	0.919	0.506	57
3D-Net	0.875	0.925	0.95	0.775	0.624	0.93	0.84	76
3D-Net-UW	0.85	0.925	0.95	0.776	0.559	0.917	0.84	76

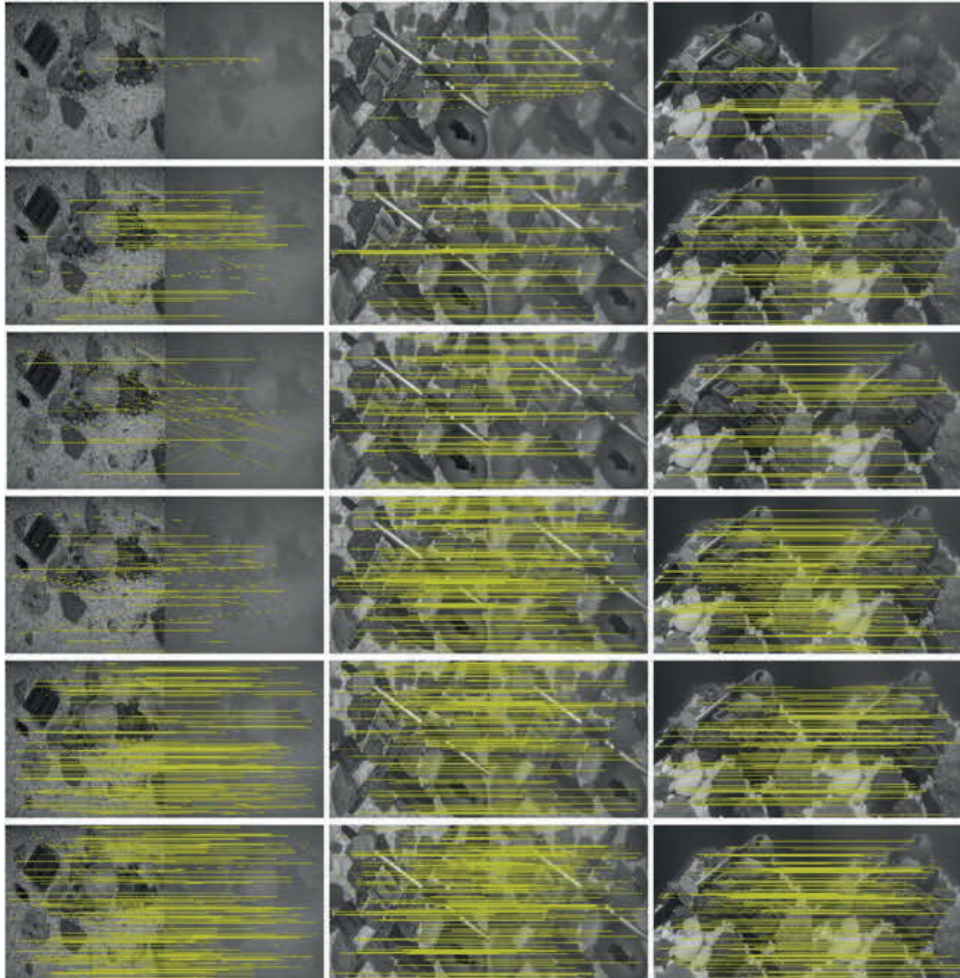


Fig. 5. Qualitative result of the interest point detection and matching. First row to last: Detection and matching with ORB(FAST+ORB), SIFT, SP – Superpoint, SP-UW – Superpoint trained with Underwater data, 3D-Net, 3D-Net-UW 3D net trained with underwater data

As an ultimate test, the proposed 3D-Net was integrated to the ORB SLAM II and conducted trajectory tracking with Aqualoc harbour [52] dataset. First, we ran the trajectories with original ORB-SLAM2 and then modified ORB SLAM2 with the proposed 3D-Net using the modified version from [29]. The resulted root mean square error (RMSE) values are provided in Table IV. Figure 6 shows generated trajectories for harbour 05 sequence. However, 3D-Net did not show a significant increase in performance, whereas trajectory H02 was not been able to track with 3D-Net compared to original ORBSLAM. For other trajectories, similar levels of performance were obtained.

TABLE IV
RMSE EVALUATION FOR AQUALOC HARBOUR DATASET

Sequence No	1	2	3	4	5	6	7
3D-Net	0.295	X	0.008	X	0.140	0.021	X
ORB-SLAM2	0.110	0.053	0.010	X	0.123	0.019	X

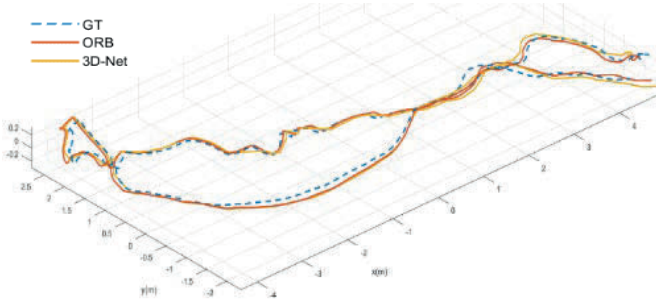


Fig. 6. Trajectories of Aqualoc harbour 05, tracking with original ORBSLAM2 and modified ORBSLAM2 with 3D-Net.

B. Evaluation of Depth Map Generation

We compared the performance of the proposed network according to [37] as it showed state-of-the-art performance.

However, [37] used RGB images as input, whereas we used greyscale images + sparse depth as input. Following error matrices [39] were employed by injection 0, 300, 500 and 1000 sparse points.

- RMSE: root mean squared error: $\sqrt{\frac{1}{T} \sum_{i=1}^T (d_i - g_i)^2}$.
- REL: mean absolute relative error: $\frac{1}{T} \sum_{i=1}^T \frac{\|d_i - g_i\|}{g_i}$.
- Mean log 10 error (log 10): $\frac{1}{T} \sum_{i=1}^T \|\log_{10} d_i - \log_{10} g_i\|$.
- Threshold accuracy: Percentage of d_i such that $\max\left(\frac{d_i}{g_i}, \frac{g_i}{d_i}\right) = \delta < \text{threshold}$.

Zero sparse points can be considered the pure depth prediction from grey images. The first set of the evaluation was conducted with an air-trained network using NYU-Depth V2. The obtained results are provided in Table V. The Junjie et al. values in Table V are extracted from [37] and tested on our platform. This result can be considered the pure depth prediction from RGB images without any sparse points, where the equivalent in the proposed is 3D-NET-0, depth prediction from grey images with zero sparse points. Comparing the first two rows from Table V shows comparative values. The resulted performance loss is due to the fact that our system uses greyscale images and a combined network, whereas [37] use RGB images with a stand-alone dedicated network. However, increasing the number of sparse depth points resulted in outstanding performance, while inputting only 300 sparse depth points resulted in a 50 % improvement compared to [37]. Figure 7 shows qualitative findings for created depth images with various sparse points. Figure 8 shows the qualitative results of depth image generation using underwater images compared to depth image generation employing [37].

TABLE V
COMPARISON OF 3D-NET FOR DEPTH PREDICTION WITH NYU-DEPTH V2. LAST ROW SHOWS RESULT FOR TUM DATASET GENERATED ON THE FLY WITH ORBSLAM2

Method	RMS	REL	log 10	$\delta < 1.25$	$\delta < 1.252$	$\delta < 1.253$
Junjie et al	0.53	0.115	0.05	0.866	0.975	0.993
3D-NET-0	0.69	0.182	0.075	0.782	0.929	0.980
3D-NET-100	0.641	0.098	0.05	0.851	0.932	0.966
3D-NET-300	0.227	0.032	0.013	0.983	0.996	0.999
3D-NET-500	0.184	0.026	0.011	0.988	0.997	0.999
3D-NET-1000	0.145	0.022	0.009	0.992	0.998	0.999
SLAM-1000	0.408	0.103	0.053	0.989	0.996	0.999

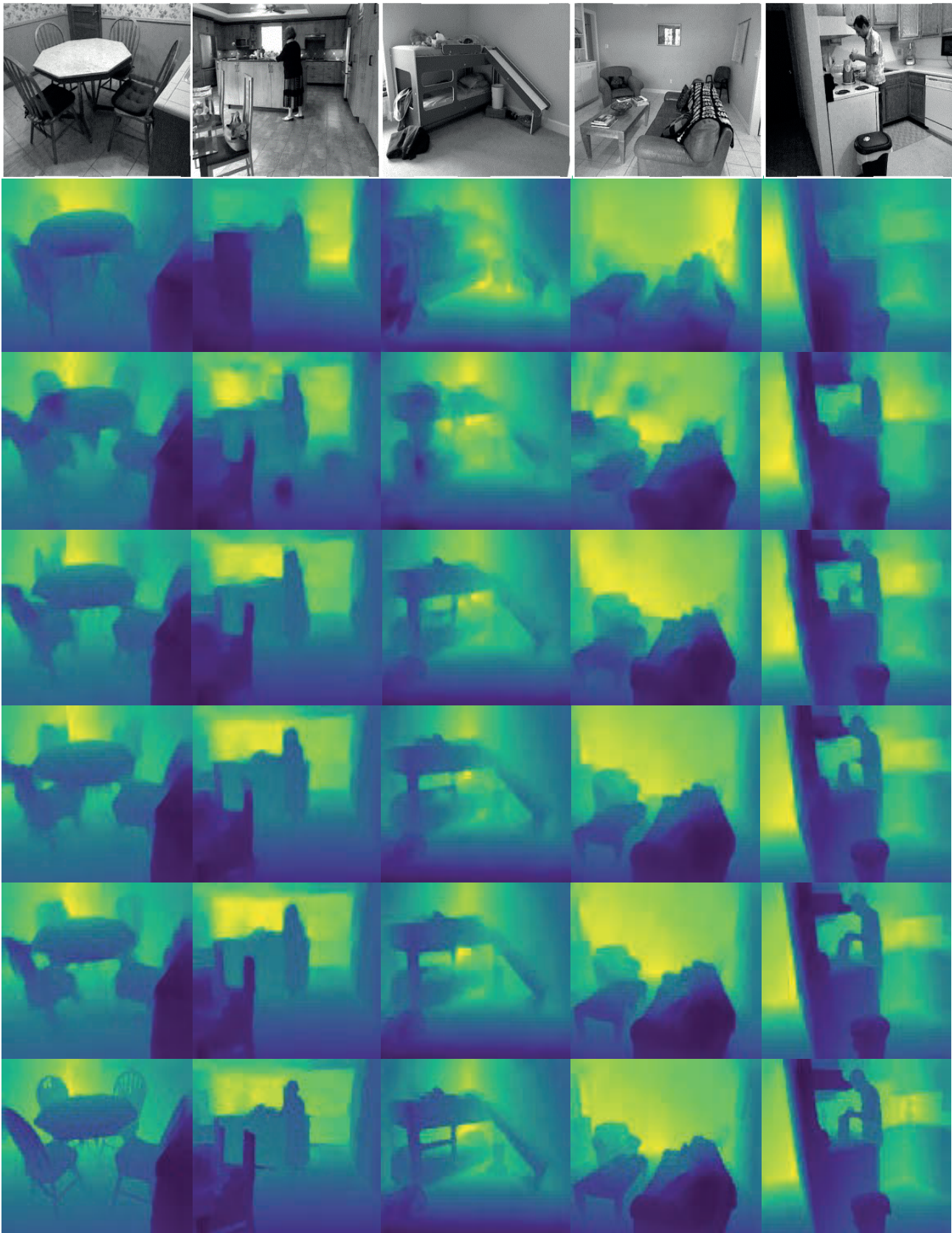


Fig. 7. Qualitative result of the depth map generation from 3D-Net, from first row to last row – the predicted depth map with 0 sparse points, 100 sparse points, 300 sparse points, 500 sparse points, 1000 sparse points, and lastly ground truth depth map.

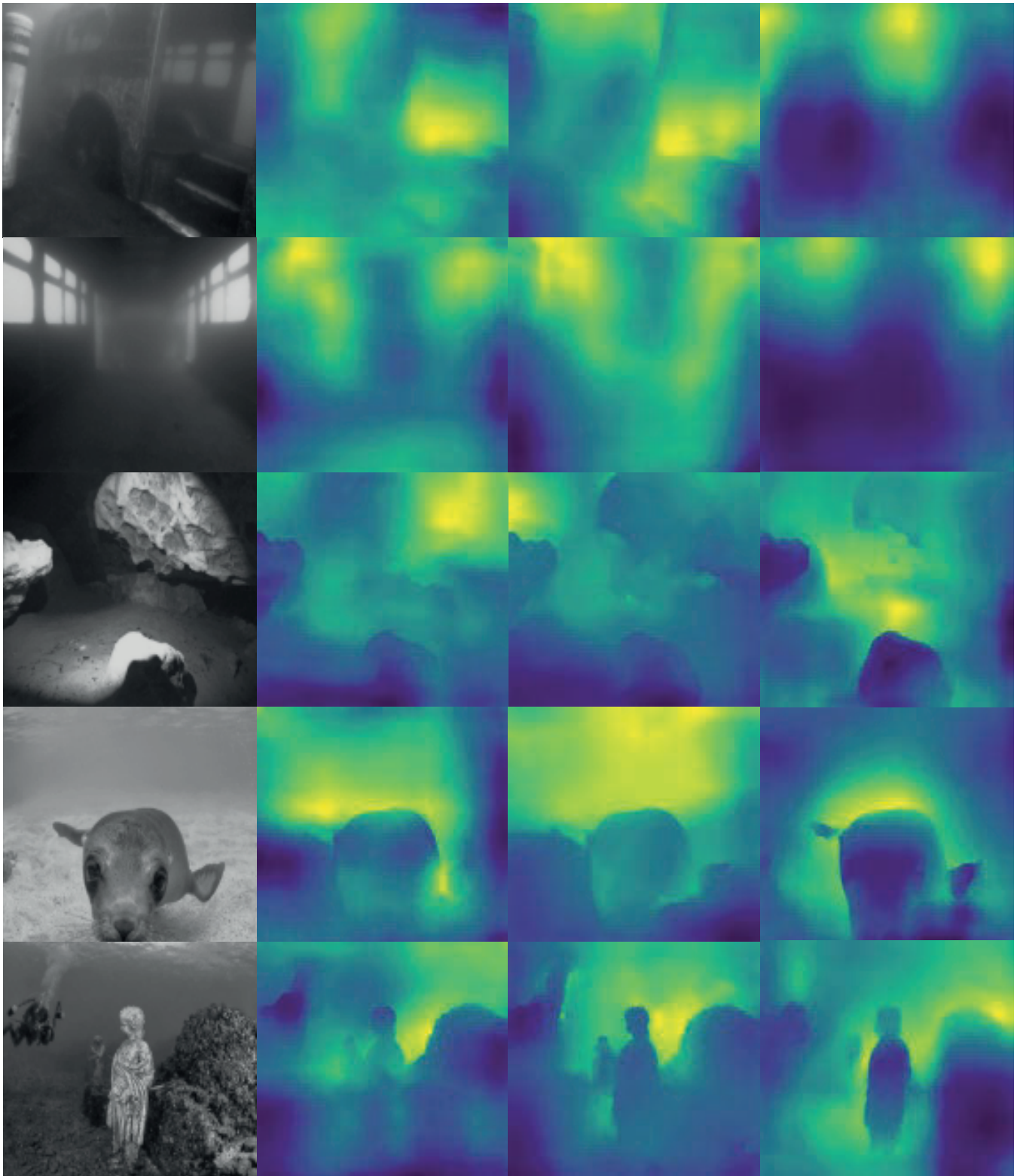


Fig. 8. Qualitative result of the depth map generation from 3D-Net in comparison with [37]. First column to last: Original greyscale image, 3D-Net trained with air data, 3D-Net trained with underwater data and depth image generated with [37].

Further, we used a SLAM system to evaluate the performance of the generated depth maps, we used ORBSLAM 2 with RGBD setup to track sequence from TUM dataset [53]. Experiments were conducted in two cases. Case 1: a new set of depth images were generated using 3D-Net injecting 0, 100,

300, 500, and 1000 sparse points from ground truth depth images and greyscale images. Generated depth images were used in the SLAM system to track the trajectories. Case 2: we modified the SLAM system to output optimized 3D map points after each bundle adjustment process. The map points serve as

the sparse depth input to the 3D-Net, along with the corresponding upcoming image. This combination enables the generation of depth images in real time. These generated depth images, along with the corresponding greyscale images, are then utilized within the SLAM system in real time. This configuration emulates a virtual depth camera, where the initial depth image is generated without any sparse points. For both cases, we used the intrinsic correction of the camera proposed in [14]. The trajectories generated in both cases were evaluated based on the RMSE. The results are provided in Table VI. In Case 2, the SLAM system was configured to detect 1000 points per image, and on average, 500 points were utilized in the sparse depth maps. The calculated RMSE value is tabulated in the column “SLAM-1000” in Table VI. The row named “SLAM-1000” in Table V illustrates the values of error

matrices described at the beginning of this section for TUM dataset. As in Case 2, a similar setup was used to generate trajectories for Aqualoc harbour 01 sequence. The resulting trajectories and RMSE values are illustrated in Fig. 9. Ground truth of the trajectory is extracted from the dataset itself.

C. Generation of Dense Depth Map

Another interesting usage of the deep learned depth image is the construction of a large dense map. Figure 1 shows the generation of a dense map for underwater scenarios. To the best of our knowledge, it has not previously been attempted as the depth cameras have a very limited range and noise measurement in underwater environments. Figure 10 shows a qualitative comparison with a dense map generated by a depth camera vs. dense map generated for TUM dataset.

TABLE VI
RMSE EVALUATION FOR TUM XYZ1 DATASET ON PREDICTED DEPTH IMAGES

Input Type	RGBD	0	100	300	500	1000	SLAM-1000
RMSE	0.050	0.157	0.117	0.072	0.064	0.056	0.053

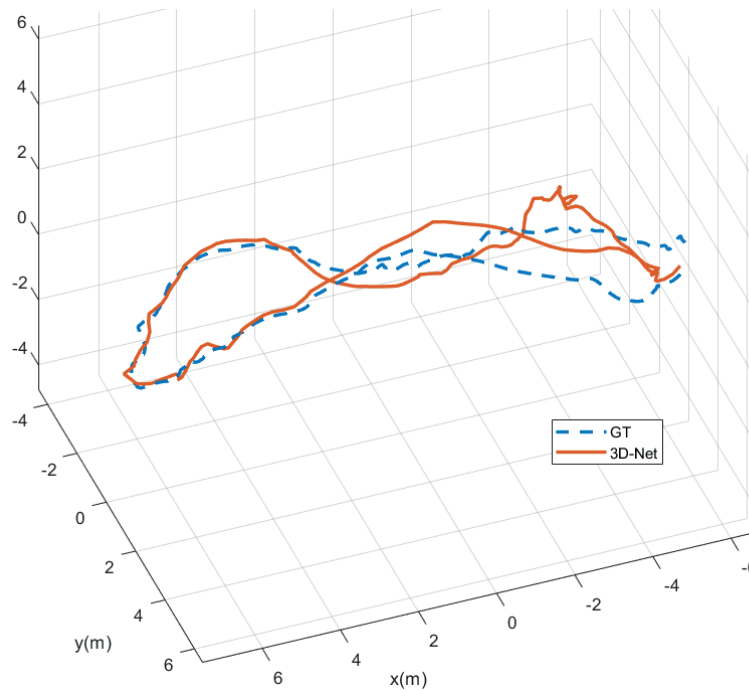


Fig. 9. Trajectories of Aqualoc harbour 01, tracking with ORBSLAM2 in RGBD setup. RMES of the generated trajectory is 0.388.

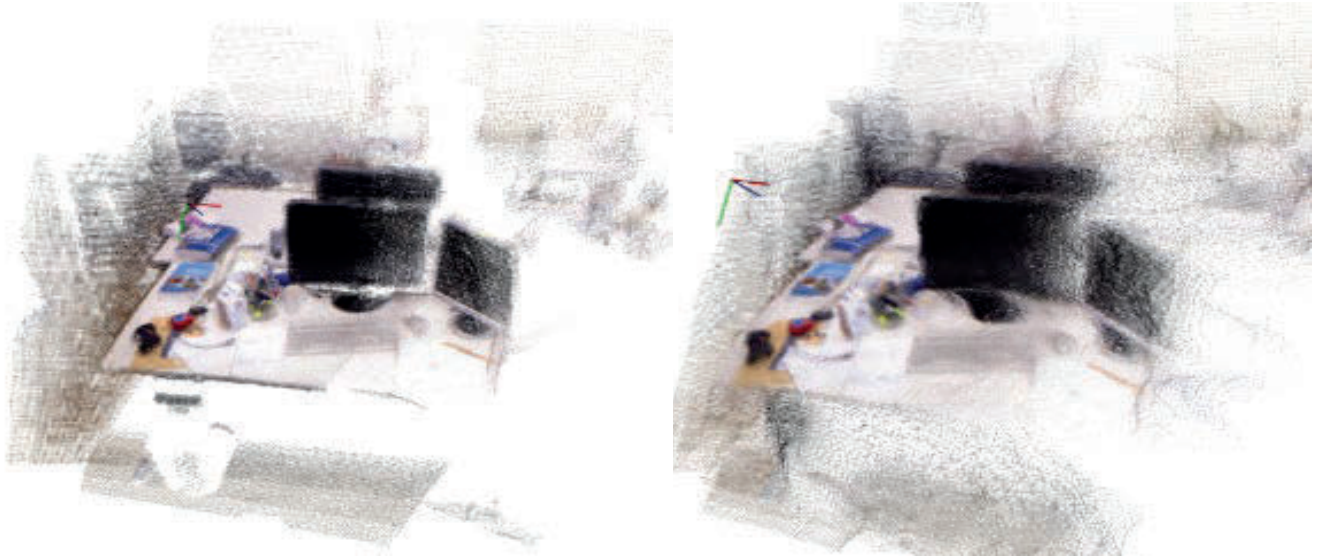


Fig. 10. Qualitative result of the depth map generation from RGBD camera (Left) 3D -Net (Right): 1000 sparse points.

VI. CONCLUSION AND FUTURE WORK

With encouraging results, deep learning has lately dominated several computer visions challenges. Deep learning approaches to the vision odometry problem are, regrettably, still in their infancy. The literature reported a very few attempts, almost all of them focusing on-ground robots and applications. There is relatively little research on VO, not even connected to 3D geometry issues, and none of them especially deals with the underwater VO. We tried to develop an assisted network to conventional VO algorithm such that the system would be able to achieve additional benefits with promising performance and accuracy compared to a classical system. We proposed a combined network with several branches out for the detection of points of interest, descriptors, and depth images. The proposed network is specifically trained for underwater scenarios with the help of a GAN network. Even though interest point detection and descriptor matching show improved results on tested datasets, the proposed 3D-Net is still lacking behind in terms of generalization. However, result in section V-1 and Table IV shows that system is able to track comparative to the original ORBSLAM2 [11] except for one case. Training the network with a real underwater dataset may improve the performance and be left for future work.

The predicted depth images have shown promising results and can be used for several applications, as shown in section V-2. It is possible to use the proposed 3D Net as a virtual depth camera and track trajectories using RGBD SLAM. However, the system has a poor performance in areas such as edges and corners, but the predicted depth values are within the tolerance, such that the SLAM optimization module is able to correct them. Further proposed system can be used in areas where depth cameras fail to estimate reliable measurements such as outdoors under the sun and in underwater environments. Moreover, the predicted depth images can be used for mapping the environment as shown in Figs. 1 and 10 and provide an encouraging outcome. The proposed approach is effectively used in areas such as underwater where a dense map was

previously not possible with feature based SLAM system. However, a predicted depth image requires several filtrations to remove outliers before being used in a practical application. Figures 1 and 10 show dense point clouds without using any such filtration, and projecting points according to camera poses. Another application of the 3D-Net is to be used to generate super-resolution for sensors such as sonar and lidar. Moreover, 3D-Net can be used to fill in missing measurements from depth cameras. Furthermore, the network can also be used to predict the metric scale for monocular SLAM systems. These areas of application will be studied in future research.

REFERENCES

- [1] W. Chen *et al.*, "An overview on visual SLAM: From tradition to semantic," *Remote Sens.*, vol. 14, no. 13, Jun. 2022, Art. no. 3010. <https://doi.org/10.3390/rs14133010>
- [2] C. Cadena *et al.*, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016. <https://doi.org/10.1109/TRO.2016.2624754>
- [3] L. Lechelek, S. Homa, R. Zrour, M. Naudin, and C. Guillemin, "A hybrid method for 3D reconstruction of MR images," *Journal of Imaging*, vol. 8, no. 4, 2022, Art. no. 103. <https://doi.org/10.3390/jimaging8040103>
- [4] K. Hu, J. Wu, Y. Li, M. Lu, L. Weng, and M. Xia, "FedGCN: Federated learning-based graph convolutional networks for non-Euclidean spatial data," *Mathematics*, vol. 10, no. 6, 2022, Art. no. 1000. <https://doi.org/10.3390/math10061000>
- [5] K. Hu, C. Weng, Y. Zhang, J. Jin, and Q. Xia, "An overview of underwater vision enhancement: From traditional methods to recent deep learning," *J. Mar. Sci. Eng.*, vol. 10, no. 2, Feb. 2022, Art. no. 241. <https://doi.org/10.3390/jmse10020241>
- [6] K. Hu, M. Li, M. Xia, and H. Lin, "Multi-scale feature aggregation network for water area segmentation," *Remote Sensing*, vol. 14, no. 1, Jan. 2022, Art. no. 206. <https://doi.org/10.3390/rs14010206>
- [7] H. M. S. Bruno and E. L. Colombari, "LIFT-SLAM: A deep-learning feature-based monocular visual SLAM method," *Neurocomputing*, vol. 455, pp. 97–110, Sep. 2021. <https://doi.org/10.1016/j.neucom.2021.05.027>
- [8] Y. Cao, Y. Luo, and T. Wang, "ORB-SLAM implementation using deep learning methods for visual odometry." [Online]. Available: https://ty-wang.github.io/data/slam_report.pdf
- [9] X. Gao and T. Zhang, "Unsupervised learning to detect loops using deep neural networks for visual SLAM system," *Auton. Robots*, vol. 41, no. 1, pp. 1–18, Dec. 2017. <https://doi.org/10.1007/s10514-015-9516-2>

- [10] J. Oh and G. Eoh, "Variational Bayesian approach to condition-invariant feature extraction for visual place recognition," *Applied Sciences*, vol. 11, no. 19, Sep. 2021, Art. no. 8976. <https://doi.org/10.3390/app11198976>
- [11] R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017. <https://doi.org/10.1109/TRO.2017.2705103>
- [12] C. Campos, R. Elvira, J. J. G. Rodriguez, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multimap SLAM," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, May 2021. <https://doi.org/10.1109/TRO.2021.3075644>
- [13] A. Steenbeek and F. Nex, "CNN-based dense monocular visual SLAM for real-time UAV exploration in emergency conditions," *Drones*, vol. 6, no. 3, Mar. 2022, Art. no. 79. <https://doi.org/10.3390/drones6030079>
- [14] K. Tateno, F. Tombari, I. Laina, and N. Navab, "CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 6565–6574. <https://doi.org/10.1109/CVPR.2017.695>
- [15] M. Bloesch, J. Czarnowski, R. Clark, S. Leutenegger, and A. J. Davison, "CodeSLAM – Learning a compact, optimisable representation for dense visual SLAM," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 2560–2568. <https://doi.org/10.1109/CVPR.2018.00271>
- [16] N. Yang, R. Wang, J. Stückler, and D. Cremers, "Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., vol. 11212. Springer, Cham, 2018, pp. 835–852. https://doi.org/10.1007/978-3-030-01237-3_50
- [17] D. Bojanic, K. Bartol, T. Pribanic, T. Petkovic, Y. D. Donoso, and J. S. Mas, "On the comparison of classic and deep keypoint detector and descriptor methods," in *Int. Symp. Image Signal Process. Anal. ISPA*, vol. 2019, Dubrovnik, Croatia, Sep. 2019, pp. 64–69. <https://doi.org/10.1109/ISPA.2019.8868792>
- [18] S. Dara and P. Tumma, "Feature extraction by using Deep Learning: A survey," in *Proc. 2nd Int. Conf. Electron. Commun. Aerosp. Technol. ICECA 2018*, Coimbatore, India, Mar. 2018, pp. 1795–1801. <https://doi.org/10.1109/ICECA.2018.8474912>
- [19] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "SVO: Semi-direct visual odometry for monocular and multi-camera systems," in *2014 IEEE Int. Conf. Robot. Autom.*, 2014, pp. 1–18. [Online]. Available: https://rpg.ifi.uzh.ch/docs/TRO16_Forster-SVO.pdf
- [20] J. Engel, V. Koltun, and D. Cremers, "Direct Sparse Odometry," 2016. [Online]. Available: <https://jakobengel.github.io/pdf/DSO.pdf>
- [21] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, Kerkira, Greece, Sep. 1999, pp. 1150–1157. <https://doi.org/10.1109/ICCV.1999.790410>
- [22] H. Bay, T. Tuytelaars, and L. Van Gool, "LNCS 3951 – SURF: Speeded up robust features," in *Computer Vision – ECCV 2006. Lecture Notes in Computer Science*, A. Leonardis, H. Bischof, and A. Pinz, Eds., vol. 3951. Springer, Berlin, Heidelberg, 2006, pp. 404–417. https://doi.org/10.1007/11744023_32
- [23] M. Calonder, V. Lepetit, M. Özuysal, T. Trzcinski, C. Strecha, and P. Fua, "BRIEF: Computing a local binary descriptor very fast," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1281–1298, 2012. <https://doi.org/10.1109/TPAMI.2011.222>
- [24] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. IEEE Int. Conf. Comput. Vis.*, Barcelona, Spain, Nov. 2011, pp. 2564–2571. <https://doi.org/10.1109/ICCV.2011.6126544>
- [25] D. Detone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-supervised interest point detection and description," in *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, Salt Lake City, UT, USA, Jun. 2018, pp. 337–349. <https://doi.org/10.1109/CVPRW.2018.00060>
- [26] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "LIFT: Learned invariant feature transform," in *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., vol. 9910 LNCS. Springer, Cham, 2016, pp. 467–483. https://doi.org/10.1007/978-3-319-46466-4_28
- [27] C. B. Choy, J. Y. Gwak, S. Savarese, and M. Chandraker, "Universal correspondence network," *Adv. Neural Inf. Process. Syst.*, pp. 2414–2422, Jun. 2016.
- [28] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer, "Discriminative learning of deep convolutional feature point descriptors," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, Dec. 2015, pp. 118–126. <https://doi.org/10.1109/ICCV.2015.22>
- [29] C. Deng, K. Qiu, R. Xiong, and C. Zhou, "Comparative study of Deep Learning based features in SLAM," in *2019 4th Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)*, Nagoya, Japan, Jul. 2019, pp. 250–254. <https://doi.org/10.1109/ACIRS.2019.8935995>
- [30] X. Han, Y. Tao, Z. Li, R. Cen, and F. Xue, "SuperPointVO: A lightweight visual odometry based on CNN feature extraction," in *2020 5th International Conference on Automation, Control and Robotics Engineering (CACRE)*, Dalian, China, Sep. 2020, pp. 685–691. <https://doi.org/10.1109/CACRE50138.2020.9230348>
- [31] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Self-improving visual odometry," *CoRR*, vol. abs/1812.03245, 2018. [Online]. Available: <http://arxiv.org/abs/1812.03245>
- [32] N. Yang, L. Von Stumberg, R. Wang, and D. Cremers, "D3VO: Deep depth, deep pose and deep uncertainty for monocular visual odometry," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Seattle, WA, USA, Jun. 2020, pp. 1278–1289. <https://doi.org/10.1109/CVPR42600.2020.00136>
- [33] H. Zhan, C. S. Weerasekera, J. W. Bian, and I. Reid, "Visual odometry revisited: What should be learnt?," in *Proc. – IEEE Int. Conf. Robot. Autom.*, Paris, France, May 2020, pp. 4203–4210. <https://doi.org/10.1109/ICRA40945.2020.9197374>
- [34] A. Ranjan et al., "Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Long Beach, CA, USA, Jun. 2019, pp. 12232–12241. <https://doi.org/10.1109/CVPR.2019.01252>
- [35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, Dec. 2016, pp. 770–778. <https://doi.org/10.1109/CVPR.2016.90>
- [36] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd Int. Conf. Learn. Represent. ICLR 2015 – Conf. Track Proc.*, 2015, pp. 1–14. <https://doi.org/10.48550/arXiv.1409.1556>
- [37] J. Hu, M. Ozay, Y. Zhang, and T. Okatani, "Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries," *Proc. – 2019 IEEE Winter Conf. Appl. Comput. Vision, WACV 2019*, Waikoloa, HI, USA, Jan. 2019, pp. 1043–1051. <https://doi.org/10.1109/WACV.2019.00116>
- [38] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *Proc. – 2016 4th Int. Conf. 3D Vision, 3DV 2016*, Stanford, CA, USA, Oct. 2016, pp. 239–248. <https://doi.org/10.1109/3DV.2016.32>
- [39] F. Mal and S. Karaman, "Sparse-to-dense: Depth prediction from sparse depth samples and a single image," in *Proc. – IEEE Int. Conf. Robot. Autom.*, Brisbane, QLD, Australia, May 2018, pp. 4796–4803. <https://doi.org/10.1109/ICRA.2018.8460184>
- [40] Y. Y. Jau, R. Zhu, H. Su, and M. Chandraker, "Deep keypoint-based camera pose estimation with geometric constraints," in *IEEE Int. Conf. Intell. Robot. Syst.*, Las Vegas, NV, USA, Oct. 2020, pp. 4950–4957. <https://doi.org/10.1109/ROS45743.2020.9341229>
- [41] T. Y. Lin et al., "Microsoft COCO: Common objects in context," in *Computer Vision – ECCV 2014. Lecture Notes in Computer Science*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., vol. 8693. Springer, Cham., 2014, pp. 740–755. https://doi.org/10.1007/978-3-319-10602-1_48
- [42] P. K. N. Silberman, D. Hoiem and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Computer Vision – ECCV 2012. Lecture Notes in Computer Science*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds., vol. 7576. Springer, Berlin, Heidelberg, 2012, pp. 746–760. https://doi.org/10.1007/978-3-642-33715-4_54
- [43] A. Dancu, M. Fourgeaud, Z. Franjic, and R. Avetisyan, "Underwater reconstruction using depth sensors," in *SA'14, SIGGRAPH Asia 2014 Tech. Briefs*, Nov. 2014, Art. no. 2, pp. 1–4. <https://doi.org/10.1145/2669024.2669042>

- [44] S. T. Digumarti, G. Chaurasia, A. Taneja, R. Siegwart, A. Thomas, and P. Beardsley, "Underwater 3D capture using a low-cost commercial depth camera," in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Lake Placid, NY, USA, 2016, pp. 1–9. <https://doi.org/10.1109/WACV.2016.7477644>
- [45] N. Wang, Y. Zhou, F. Han, H. Zhu, and Y. Zheng, "UWGAN: Underwater GAN for real-world underwater color restoration and dehazing," *arXiv* 2019, arXiv:1912.10269, pp. 1–10, 2019. <https://arxiv.org/ftp/arxiv/papers/1912/1912.10269.pdf>
- [46] M. J. Islam, Y. Xia, and J. Sattar, "Fast underwater image enhancement for improved visual perception," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 3227–3234, Feb. 2020. <https://doi.org/10.1109/LRA.2020.2974710>
- [47] C. Fabbri, M. J. Islam, and J. Sattar, "Enhancing underwater imagery using generative adversarial networks," in *Proc. - IEEE Int. Conf. Robot. Autom.*, Brisbane, QLD, Australia, May 2018, pp. 7159–7165. <https://doi.org/10.1109/ICRA.2018.8460552>
- [48] M. Trajković and M. Hedley, "Fast corner detection," *Image Vis. Comput.*, vol. 16, no. 2, pp. 75–87, Feb. 1998. [https://doi.org/10.1016/S0262-8856\(97\)00056-5](https://doi.org/10.1016/S0262-8856(97)00056-5)
- [49] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk, "HPatches: A benchmark and evaluation of handcrafted and learned local descriptors," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 3852–3861. <https://doi.org/10.1109/CVPR.2017.410>
- [50] A. Duarte, F. Codevilla, J. D. O. Gaya, and S. S. C. Botelho, "A dataset to evaluate underwater image restoration methods," in *OCEANS 2016 – Shanghai*, Shanghai, China, Apr. 2016, pp. 1–6. <https://doi.org/10.1109/OCEANSAP.2016.7485524>
- [51] C. Li *et al.*, "An underwater image enhancement benchmark dataset and beyond," *IEEE Trans. Image Process.*, vol. 29, pp. 4376–4389, Nov. 2020. <https://doi.org/10.1109/TIP.2019.2955241>
- [52] M. Ferrera, V. Creuze, J. Moras, and P. Trouvé-Peloux, "AQUALOC: An underwater dataset for visual-inertial-pressure localization," *Int. J. Rob. Res.*, vol. 38, no. 14, pp. 1549–1559, Oct. 2019. <https://doi.org/10.1177/0278364919883346>
- [53] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura-Algarve, Portugal, Oct. 2012, pp. 573–580. <https://doi.org/10.1109/IROS.2012.6385773>



Chinthaka Amarasinghe has received his B.Sc. (honours) specializing in the field of Mechanical Engineering, from University of Moratuwa, Sri Lanka, in 2010. He is a full-time Lecturer at Uva Wellassa University attached to the Department of Science and Technology. His research interest includes robot navigation, artificial intelligence and machine learning.

E-mail: chinthakaa@uwu.ac.lk

ORCID iD: <https://orcid.org/0000-0003-2049-5584>



Asanaga Rathnaweera has obtained his B.Sc. (honors) specializing in the field of Production Engineering, the University of Peradeniya in 1997. He obtained his PhD in Mechanical Engineering from the University of Melbourne, Australia in 2004. He has been serving for nearly 25 years at the University of Peradeniya as an academic staff member in various capacities. Further, Prof. Rathnaweera worked as an adjunct professor in Mechanical Engineering at Texas Tech University, USA from 2010 – 2011 on sabbatical leave. His major research interests include artificial intelligence, optimization, image processing, automobile engineering, and machining technology.

E-mail: asanga@eng.pdn.ac.lk

ORCID iD: <https://orcid.org/0000-0002-8600-3418>



Sanjeeva Maitripala has started his career as a Lecturer at the Department of Mechanical and Manufacturing Engineering, University of Ruhuna, Galle, Sri Lanka, in 2006. Since 2009 he has been serving as a Senior Lecturer of the Department of Mechanical Engineering, University of Peradeniya, Sri Lanka. He has also served as a visiting research scholar at the Nanotech Center of Texas Tech University on several occasions and currently serves as a visiting Associate Professor of the Systems and Controls Department of IIT-Bombay. He has so far received four presidential awards in recognition for high impact scientific publications in Sri Lanka and his research interests include decision making theory and nonlinear geometric control with a focus on autonomous navigation and control of unmanned vehicles.

E-mail: smaitthri@eng.pdn.ac.lk