Deep Learning-Enhanced Visual Monitoring in Hazardous Underwater Environments with a Swarm of Micro-Robots

Shuang Chen, Yifeng He, Barry Lennox, Farshad Arvin and Amir Atapour-Abarghouei

***Abstract*— Long-term monitoring and exploration of extreme environments, such as underwater storage facilities, is costly, labor-intensive, and hazardous. Automating this process with low-cost, collaborative robots can greatly improve efficiency. These robots capture images from different positions, which must be processed simultaneously to create a spatio-temporal model of the facility. In this paper, we propose a novel approach that integrates data simulation, a multi-modal deep learning network for coordinate prediction, and image reassembly to address the challenges posed by environmental disturbances causing drift and rotation in the robots' positions and orientations. Our approach enhances the precision of alignment in noisy environments by integrating visual information from snapshots, global positional context from masks, and noisy coordinates. We validate our method through extensive experiments using synthetic data that simulate real-world robotic operations in underwater settings. The results demonstrate very high coordinate prediction accuracy and plausible image assembly, indicating the real-world applicability of our approach. The assembled images provide clear and coherent views of the underwater environment for effective monitoring and inspection, showcasing the potential for broader use in extreme settings, further contributing to improved safety, efficiency, and cost reduction in hazardous field monitoring.**

I. Introduction

Monitoring and measuring environmental conditions are essential in extreme environments, such as those characterised by high temperatures, radiation or underwater settings. Deploying human operators in such challenging conditions is often impractical or poses significant risks. Consequently, robotic systems present a safer and more reliable alternative for conducting these critical missions. For instance, in nuclear power plants, nuclear waste is stored in pools of water known as spent fuel ponds contained within specially designed rods. To ensure the safety and integrity of these storage sites, specialised underwater camera systems monitor the condition, position, and quantity of the nuclear waste. The International Atomic Energy Agency (IAEA) allocates substantial resources—over £25 million annually [1] to inspect nuclear fuel waste storage using the manual IAEA DCM-14 camera [2]. However, regularly inspecting the ponds is time-consuming and repetitive, highlighting the need for more efficient monitoring solutions, such as automated robotic systems.

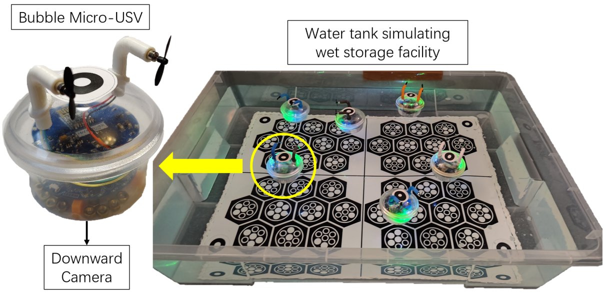

Fig. 1. The overarching vision of the proposed visual exploration system involves deploying a swarm of micro-surface robots, *Bubbles*.

While robots with specialised sensory systems can automate inspection, relying on a single-robot configuration, in large and uncharted environments, can be a significant concern. Consequently, deploying multiple robots can enhance coverage, robustness, and effectiveness in these settings [3].

To enable a full inspection of the scene monitored by underwater robots, the snapshots captured the robots must be assembled into a cohesive view. However, this task is complicated by the nature of the environment. The movement of the liquid in the pond causes the robots to drift unpredictably, which introduces random and irregular noise into positional coordinates and rotation angle information recorded during snapshot capture. As a result, attempts to directly stitch these noisy snapshots produce confusing and misaligned images, which do not provide a clear understanding of the scene (as the noisy stitched image shown in Fig. 5).

In this paper, we propose a pipeline that includes data simulation, a multi-modal deep learning network for coordinate prediction, and image reassembly. A novel approach is introduced for generating synthetic images, that simulate spent fuel ponds, described in Sec. IV-A. Additionally, we present a dataset SFP10 with 11,000 sets of disturbed data, simulating the drifting effects experienced by robots navigating the fuel pond. The code and dataset are publicly available.

The contributions in our work are summarised as:

- **Robotics:** We propose a robust, low-cost methodology for coordinating robotic swarm in underwater environments, ensuring effective image capture and assembly despite disturbances such as drifting and rotation.
- **Pipeline:** We introduce an integrated pipeline that combines data simulation, coordinate prediction through a multi-modal deep learning network, and robust image

This work was partially supported by the UK EPSRC RAIN, Robotics & Artificial Intelligence for Nuclear [grant numbers EP/R026084/1 and EP/W001128/1].

S. Chen, F. Arvin and A. Atapour-Abarghouei are with the Department of Computer Science, Durham University, Durham, UK shuang.chen@durham.ac.uk

Y. He and B. Lennox are with the Department of Electrical & Electronic Engineering, The University of Manchester, Manchester, UK.

Code and dataset can be found at: https://github.com/ChrisChen1023/Micro-Robot-Swarm assembly, significantly improving the precision of alignment in noisy environments.

- **Dataset:** We create and release a comprehensive dataset SFP10 with 11,000 sets of disturbed data to simulate real-world robotic operations in challenging underwater scenes, providing a valuable resource for future research.

II. Related Work

We consider related work in Robotic Visual Monitoring (Sec. II-A) and Image Assembly (Sec. II-B).

A. Visual Monitoring by Micro Robots

While prior work [4], [5], [6] has focused on robotic inspection of storage facilities, the implementation of such automated systems presents significant challenges. For instance, [7] developed *MallARD*, an autonomous aquatic surface vehicle specifically designed for monitoring nuclear storage ponds. MallARD is equipped with four thrusters that enable navigation across the two-dimensional water surface and includes a camera in its payload area for conducting underwater inspections. Similarly, the *MASKI+* robotic system [8] uses a controller that provides five degrees of freedom in water for diagnostic and intervention tasks in hydroelectric power plants. We also developed *AVEXIS* [9], a low-cost micro-submersible designed to monitor nuclear underwater storage facilities with limited access points. In a subsequent deployment [10], we equipped the robot with radiological sensors for inspecting the Fukushima Daiichi site in Japan.

For multi-robot examples, [11] introduced a multi-robot coverage problem in a barrier-laden environment using the Pioneer P3-DX robot platform and a simulated barrier coverage robot. Similarly, in [12], multiple robots were developed for inspecting oil and gas pipelines through a wireless autonomous surface vehicle (ASV) relay, enabling coordinated operations. [13] employed an ASV named *Aqua2* for underwater exploration, specifically for mapping and monitoring shipwrecks. Despite the results of prior work, generating an accurate long-term spatio-temporal model of these facilities remains the main challenge that needs to be addressed. Our approach can assemble snapshots captured by micro robot swarms to allow for a complete live inspection of the facility.

B. Image Reassembly

Recent advances in image reassembly (reassembling images from disjointed fragments) have explored unsupervised and self-supervised learning approaches. For instance, [14] integrates geometry and colour information for image reassembly by pairwise matching of fragment boundaries, global image reassembly through a graph-based search algorithm and refinement of the reassembly using a graph optimisation technique [15] to reduce accumulated errors. Learning-based approaches have also been used for image reassembly. For example, [16] integrates both boundary and semantic information to improve puzzle reconstruction. This multi-task pipeline incorporates a branch for predicting jigsaw permutations and another branch for generating images [17] with the correct order. [18] combines reinforcement learning [19] with Siamese networks [20] to optimise fragment swapping to correctly reassemble puzzles.

Image assembly and puzzle solving methods have also played a significant role in enhancing the performance of other tasks. [21] introduces a self-supervised approach for learning image representations by solving jigsaw puzzles as a pretext task, to allow the network to learn object parts and their spatial arrangement. Puzzle-solving [22] has also been used to enhance anomaly detection in images. JiGen [23] leverages jigsaw puzzle-solving for domain generalisation by jointly learning spatial correlations and object classification. [24] uses weak spatial cues to iteratively solve jigsaw puzzles by combining unary and binary terms to assess the likelihood of patches being correctly positioned relative to one another.

Despite the advances in image reassembly techniques, none of these methods are equipped to handle the unique challenges of rotation and overlap that appear in our application. In the context of our work, fluid dynamics may cause the robots to move and rotate unpredictably. The precise amount of rotation before each snapshot is captured can vary, as the robots are subject to water currents, buoyancy, and other forces. This makes it challenging to reconstruct images when the degree of rotation is uncertain and not accounted for. Additionally, many of the snapshot overlap, complicating the alignment process. Many existing methods [14], [16] assume fixed orientations and non-overlapping images and are, as such, unsuitable in our application. Consequently, we have developed a novel pipeline to address these specific challenges to ensure robust image reassembly even with rotational variation and overlapping image regions.

III. Micro-Robotic Platform

In this section, we briefly introduce main components of the *Bubble* micro surface robot which has been developed for perpetual monitoring of wet storage facilities [25]. Figure 2 illustrates robot's hardware module and architecture.

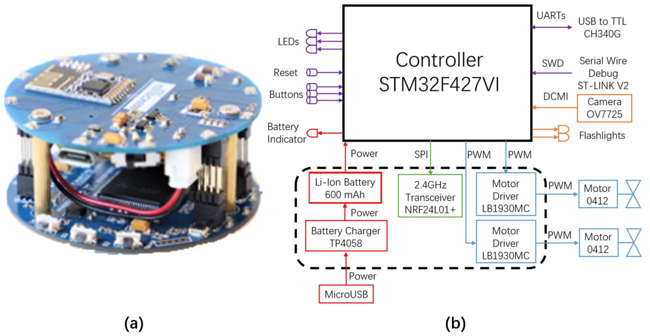

Fig. 2. (a) The electronics of Bubble, divided into two PCBs with different functionalities. (b) The system architecture of the robot illustrates its key functions. The section within the dashed rectangle represents the top PCB.

A. Bubble Robot Specification

An STM32F427VI microcontroller manages low-level tasks, such as image capture, actuation, and communication. The low-power microcontroller is a Cortex-M4 single-core 32-bit processor operating at 180 MHz with 0.25 MB SRAM and 2 MB Flash memory embedded with a Floating Processing Unit (FPU), which enables the real-time image processing. An NRF24L01+ wireless communication module is attached to the main board of Bubble to transfer images to the base station. The robot uses a compact low-cost OV7725 camera with a maximum resolution of 640×480 pixels at 30 fps. 160×110-pixel images are used here to optimise memory usage and communication bandwidth. The captured image goes into an external buffer, but the robot can store eight images locally. This can be extended by reducing the image size and increasing the memory size.

As shown in Fig. 1, the Bubble enclosure is 3D printed out of plastic. Two actuators on top reduce cross contamination from storage facilities. The enclosure is a waterproof transparent case allowing camera to see downward and capture images from the bottom of the pond. The Bubble is equipped with two coreless DC motors as its actuators. The motors are paired with 2.1 cm diameter twin-wing plastic propellers commonly used in tiny drones.

B. Experimental Setup

The experimental simulation pond is 75×52 cm. Each robotic unit is equipped with circular markers, with a diameter of 3 cm each. The hexagonal patterns and the solid and hollow circles on the poster at the bottom of the simulation pond (Fig. 1) are designed to replicate the layout and features of an actual spent fuel pond in nuclear power stations.

An efficient swarm coordination mechanism is essential for conducting such exploration tasks. Collective Motion, a standard swarm behaviour, offers a promising approach for real-world applications by enabling coordinated movement and operation of multiple agents. In this work, we used the state-of-the-art collective motion mechanism that is based on the active elastic sheet framework [26]. It places virtual springs between adjacent robots, where both distance and angular differences between robots generate repulsive and attractive forces to enable the robots to remain aligned and achieve coordinated collective motion.

Whycon [27] is used with a low-cost USB camera to localise and track the robots. The camera captures the entire area from an overhead perspective. Each Bubble robot is equipped with a circular marker to enable tracking. Due to hardware limitations, only a single Bubble robot can communicate simultaneously with the other six Bubbles. In this configuration, one robot acts as the leader, while the other six serve as followers.

IV. Dataset

Since the learning-based model in charge of predicting correct robot coordinates requires a significant amount of data to be trained, capturing real-world data using the simulation pond is not feasible. As such, we require synthetic pond images, snapshot masks and noisy coordinates of the robot to train and evaluate our model. The model trained on the synthetic data will predict accurate coordinates by combining visual information from snapshots, spatial context from masks, and positional data from noisy coordinates.

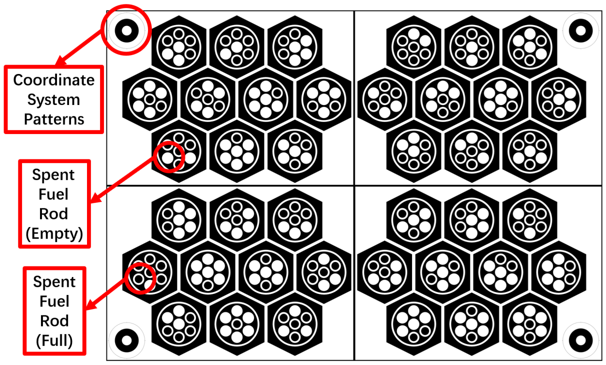

Fig. 3. Image simulating the spent fuel pond; hollow circles indicate empty rods and solid circles indicate full rods.

In this work, we present a synthetic dataset, SFP10, designed to simulate data captured from spent fuel ponds. The dataset consists of 10,000 images for training and 1,000 for testing. Each set includes synthetic pond images, snapshots, perturbed positional coordinates as well as their corresponding binary masks. The synthetic pond images replicate the layout of a spent fuel pond, while snapshots capture various robot viewpoints and orientations as they navigate the environment. The perturbed positional coordinates are added to simulate real-world drift and rotational noise, which are inherent in such environments. The binary masks provide spatial context by explicitly marking the regions covered by the snapshots, facilitating accurate prediction of the robots' positions and orientations. These components are detailed in the following.

A. Pond Image Synthesis

Each generated image represents a top-down view of fuel rods submerged in water, with variations in the internal structure of the rods (as shown in Fig. 3). Specifically, the inner circles of each rod are either white or partially filled. We assign a 0.5 probability for each inner circle to be fully or partially filled, while other characteristics, such as size, arrangement, and position of the rods, remaining fixed. Using this method, we generate 100 base images for use.

B. Snapshot Acquisition

After generating the base pond images, we simulate the capture process carried out by the swarm of micro-surface robots. The Bubbles are designed to move across the pond and capture images from various positions (x,y) and orientations (θ), representing their rotational angle. From each of the 100 base images, we randomly sample 221 snapshots with unique (x,y,θ) values, simulating different viewpoints and rotations. Each snapshot is set to a fixed size of 160×110 pixels, matching the resolution in the physical setup. This process produces a diverse set of snapshots, mimicking the behaviour of the robots as they move around the pond.

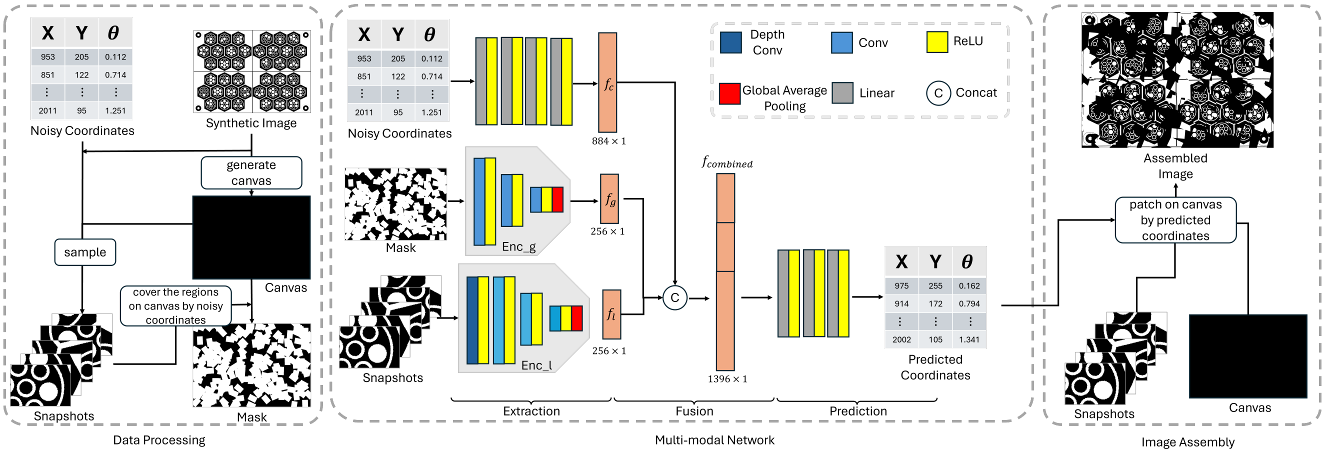

Fig. 4. Overview of the overall pipeline, which integrates data simulation (Left), noisy coordinate correction via a deep learning network (Middle), and image reassembly (Right) to generate coherent images from snapshots captured by micro-robots in noisy underwater environments.

C. *Perturbed Coordinate Data*

To simulate real-world imperfections during image capture, we introduce noise into the data. Noise is sampled from a normal distribution and added to x, y and θ, representing the effects of drifting and random rotation that could occur as the robots navigate the pond. This step simulated the small perturbations that might occur due to environmental factors or mechanical drift in the robots' movement. For each base image, we apply 100 different noise instances, for a total dataset of 10,000 sets of coordinates.

D. *Region Mapping with Mask*

To create the masks for our dataset, we generate a blank canvas with the same dimensions as the synthetic images. The snapshots are then patched onto the canvas, but instead of capturing detailed visual information, we focus on identifying the patched and unpatched regions. The areas corresponding to the snapshots are marked in white (patched), while the rest of the canvas remains black (unpatched). This binary mask explicitly represents where the snapshots are located on the image and captures their spatial orientation and rotation (Mask figure shown in Fig. 5). This global positional information is crucial for the network to understand the overall layout and context of the snapshots in relation to the entire scene.

V. Multi-Modals Image Assembly System

The pipeline (Fig. 4) consists of a data processing stage, a deep learning network for coordinate prediction, and an image stitching process to produce the final image.

A. *Problem Formulation:*

Our network aims to predict coordinates $C_{predicted}$ from noisy coordinates C_{noisy} captured by robots, where noise is introduced by environmental factors such as drift and random rotation. $C_{predicted}$ is expected to be as close as possible to C_{true}. To achieve this, we propose a multi-modal framework that integrates visual data and coordinate information to map noisy coordinates to the true ones. Given noisy coordinates $C_{noisy} = \{x_{noisy}, y_{noisy}, \theta_{noisy}\}$, corresponding snapshots S (Sec. IV-B), and binary masks M (Sec. IV-D), the task is to predict the accurate coordinate $C_{predicted} = \{x_{true}, y_{ture}, \theta_{true}\}$. This is formulated as a regression problem where the goal is to minimise the error between the predicted and true coordinates. The learned function f (our model) is expressed as:

$$C_{predicted} = f(C_{noisy}, S, M). \tag{1}$$

B. *Multi-Modal Architecture:*

The proposed network architecture is designed to handle multi-modal input, integrating information from snapshots, masks, and noisy coordinates to predict accurate, noise-free coordinates. Snapshots provide detailed local visual information, while masks offer global positional context, both of which are essential for enhancing the representation ability to predict coordinates effectively. We introduce our network in three stages: Feature Extraction, Feature Fusion, and Coordinate Prediction.

Feature Extraction: Given the input of snapshots S, masks M and coordinates C_{noisy}, we construct two encoders, Enc_l and Enc_g for S and M, respectively. Enc_l applies a depth-wise convolution [28] followed by a ReLU layer to embed S. This design choice is motivated by the nature of the snapshots: after randomly sampling 221 snapshots, each one can have a different pattern due to positional and rotational variations, making them anisotropic with respect to each other. In these cases, conventional convolution operations — where kernels process features across all input channels — are not optimal, as they risk blending distinct patterns between snapshots, failing to capture their unique characteristics. Instead, depth-wise convolutions apply separate filters to each channel independently, preserving the individuality of each snapshot's features while maintaining computational efficiency. Following the depth-wise embedding, our encoder downsamples the features for the fusion step.

For Enc_g, which processes the mask M, we apply three $\{Conv - ReLU\}$ blocks. The first one performs feature embedding, while the second and third blocks apply down-

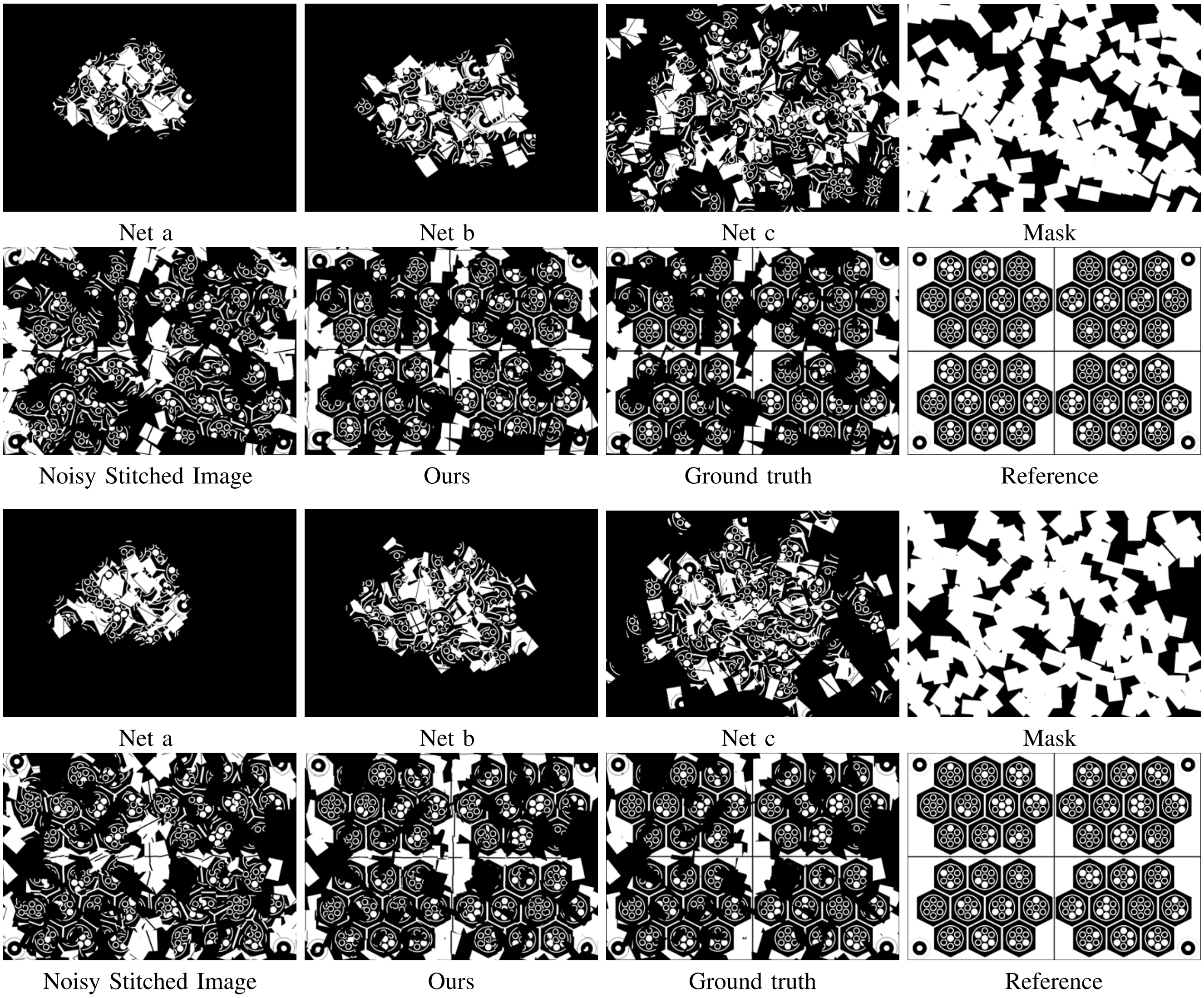

Fig. 5. The first two rows correspond to the first example, and the bottom two rows correspond to the second example. The visual comparisons demonstrate that our results exhibit a more coherent and consistent structure.

sampling to progressively reduce the spatial resolution while retaining the global information from the visual features. At the end of both encoders, global average pooling is applied to reduce the dimensionality of the visual features for a compact representation, which is appropriate given that the target output C is relatively low-dimensional. After global average pooling, we have the features f_l and f_g extracted from Enc_l and Enc_g, respectively.

For the noisy coordinates C_{noisy}, we stack four fully connected linear layers, each activated by ReLU, to embed the coordinate features into a higher-dimensional space. The embedded feature is denoted as f_c.

Feature Fusion and Coordinate Prediction: Once we extract f_l, f_g and f_c, we fuse them to create a unified feature representation $f_{combined}$. This fused representation integrates local visual information, global positional data, and the embedded noisy coordinates, providing a rich feature space that enhances the representation learning capability of the model, leading to more accurate coordinate prediction. Finally, $f_{combined}$ is passed through three fully connected layers, each activated by ReLU, to progressively refine the fused features. The network outputs final coordinates $\{$x,y,$\theta\}$.

C. Image Assembly

To assemble the final image, we process each snapshot by applying rotation, positional alignment, and compositing onto a larger canvas. Based on the provided coordinates $\{x, y, \theta\}$, the snapshot is then rotated to correct for any misalignment. Spatial alignment is achieved by positioning the rotated snapshot at the predicted coordinates by compositing the snapshot onto the canvas.

VI. Experiments

All experiments are conducted on SFP10 (Sec. IV).

A. Implementation Details

All data processing tasks - i.e. generating synthetic images, snapshots, masks, and noisy coordinates, were conducted on an Intel Core i9 10 Core Processor i9-10900X (3.7GHz) 19.25MB Cache. For model training and testing, all experiments were carried out on a single Nvidia 3070Ti GPU.

TABLE I

ABLATION STUDIES OF EACH COMPONENT.FOR MSE, MAE AND R^2, THE VALUES SEPARATED BY '/' CORRESPOND TO x, y, AND θ, RESPECTIVELY.

Net	Components				Metrics			
	Coor	Snapshots	Mask	Noisy Stitched Image	MSE↓	MAE↓	R^2 ↑	IoU↑
(a)	✓				0.0897 / 0.0901 / 0.3639	0.2514 / 0.2520 / 0.5092	-0.1047 / -0.1215 / -0.0821	18.94%
(b)	✓	✓			0.0807 / 0.0807 / 0.3328	0.2395 / 0.2405 / 0.4878	0.0060 / -0.005 / 0.0115	20.32%
(c)	✓	✓		✓	0.1384 / 0.1243 / 0.5250	0.3007 / 0.2874 / 0.5870	-0.7023 / -0.5527 / -0.5624	43.47%
Ours	✓	✓	✓		**0.0019 / 0.0022 / 0.0091**	**0.0156 / 0.0163 / 0.0329**	**0.9503 / 0.9411 / 0.9465**	**87.97%**

During training, we calculate the loss for both the location (x, y) and orientation θ using Mean Squared Error (MSE). The total loss is defined as:

$$L_{total}(C_{predicted}, C_{true}) = L_{x,y} + 0.05L_{\theta}. \qquad (2)$$

We used the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate was set to 1×10^{-4}, with a batch size of 24 to balance memory efficiency and model convergence.

B. Evaluation Metrics

To evaluate the accuracy of the predicted coordinates, we use Mean Squared Error (MSE), Mean Absolute Error (MAE), and the coefficient of determination (R^2). MSE measures the average squared difference between predicted and true coordinates, penalising larger errors more heavily, which is ideal for detecting significant deviations. MAE, on the other hand, calculates the average absolute error, and offers a straightforward interpretation of the overall accuracy. R^2 assesses how well the model's predictions fit the true coordinates. We also considered the issue of angle rotations differing by integer multiples of 360 degrees, which could lead to identical rotations but distort error measurement. However, in our experiment, the rotation of all snapshots is limited to less than 2.5 radians, which simulates real-world scenarios from our observation. Therefore, the above metrics are effective in evaluating our method.

To evaluate our assembled images, we use Intersection over Union (IoU) [29] to assess the overlap between the mask with the ground truth coordinates and the mask with the predicted coordinates. While IoU does evaluate denoising performance, it is not a precise indicator. Even with noisy coordinates, the IoU can be high (above 80%) because the noisy snapshots still overlap significantly with the ground truth region. As the model reduces noise, the IoU increases, ideally reaching 100% when all noise is removed. Therefore, while IoU helps track improvement, it must be interpreted alongside other metrics for a more accurate evaluation.

C. Evaluation and Discussion

To evaluate each component in our proposed method, we conduct four configurations: Net (a), (b), (c) and our method. The key difference between these configurations lies in their input components. Net (a) uses only [*coordinates*], while Net (b) incorporates [*coordinates, snapshots*]. Net (c) inputs [*coordinates, snapshots, noisy assembled image*]. Our proposed method employs [*coordinates, snapshots, and masks*].

As shown in Table I, Net (a) and Net (b) demonstrate that simply using noisy coordinates or noisy coordinates with snapshots fails to provide sufficient information to accurately predict the correct coordinates, resulting in poor assembly of snapshots. In contrast, our method integrates input noisy coordinates, snapshots, for local visual information, and masks, for global context, to accurately predict $\{x, y, \theta\}$, resulting in well-aligned outputs. As shown in Table I, incorporating the mask provides the global positional information necessary for the model to outperform other configurations, leading to better alignment and overall higher accuracy.

Net (c) explores the feasibility of using the noisy assembled image (with noisy coordinates) as an alternative to the mask for providing global information (replacing the mask as input for Enc_g). Even though it tends to predict the sparse $\{x, y\}$ location, it is unable to learn the global context, resulting in misalignment in the final assembled image (shown in the Fig. 5). This limitation could stems from redundant and inaccurate visual information provided by the noisy assembled image, which makes it challenging for the model to predict the θ accurately.

VII. CONCLUSIONS

In this work, we have proposed a synthetic dataset SFP10, and a novel approach for improving visual monitoring in noisy underwater environments using a swarm of microrobots. Our proposed pipeline integrates data simulation, a multi-modal deep learning network for coordinate prediction, and image reassembly to address challenges posed by environmental disturbances, such as drift and rotation. By incorporating local visual information from snapshots and global positional context from masks, our method significantly enhances the precision of both coordinate prediction and image alignment.

Although our approach has shown promising results with synthetic data, there remains a domain gap between real-world and simulated data. Future work will focus on closing this gap by exploring techniques to minimise discrepancies between synthetic data and real-world scenes. This includes exploring advanced techniques such as domain adaptation [30], transfer learning [31] and data augmentation [32] to better generalise from synthetic to real-world data. Additionally, incorporating real-world datasets into the training process [33] and leveraging unsupervised [34] or semi-supervised [35] learning approaches may further bridge the domain gap. This will ensure the robustness and applicability of our method in real-world underwater environments. Additionally, we plan to extend the system's scalability and optimise it for real-time performance [36] to broaden its use in other extreme environments such as pipeline inspections.

References

[1] S. Pepper, M. Farnitano, J. Carelli, J. Hazeltine, and D. Bailey, "Lessons learned in testing of safeguards equipment.," tech. rep., Brookhaven National Lab., Upton, NY (US), 2001.

[2] J. Doyle, *Nuclear safeguards, security and nonproliferation: achieving security with technology and policy*. Elsevier, 2011.

[3] M. Dorigo, G. Theraulaz, and V. Trianni, "Reflections on the future of swarm robotics," *Science Robotics*, vol. 5, no. 49, p. eabe4385, 2020.

[4] C. Lennox, K. Groves, V. Hondru, F. Arvin, K. Gornicki, and B. Lennox, "Embodiment of an aquatic surface vehicle in an omnidirectional ground robot," in *IEEE International Conference on Mechatronics*, vol. 1, pp. 182–186, 2019.

[5] X. Huang, F. Arvin, C. West, S. Watson, and B. Lennox, "Exploration in extreme environments with swarm robotic system," in *IEEE International Conference on Mechatronics*, vol. 1, pp. 193–198, 2019.

[6] C. West, F. Arvin, W. Cheah, A. West, S. Watson, M. Giuliani, and B. Lennox, "A debris clearance robot for extreme environments," in *Annual Conference Towards Autonomous Robotic Systems*, pp. 148–159, 2019.

[7] K. Groves, A. West, K. Gornicki, S. Watson, J. Carrasco, and B. Lennox, "Mallard: An autonomous aquatic surface vehicle for inspection and monitoring of wet nuclear storage facilities," *Robotics*, vol. 8, no. 2, p. 47, 2019.

[8] L. Provencher and S. Sarraillon, "The maski+ underwater inspection robot: A new generation ahead," in *International Conference on Applied Robotics for the Power Industry*, pp. 1–6, 2016.

[9] A. Griffiths, A. Dikarev, P. R. Green, B. Lennox, X. Poteau, and S. Watson, "Avexis—aqua vehicle explorer for in-situ sensing," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 282–287, 2016.

[10] M. Nancekievill, A. Jones, M. Joyce, B. Lennox, S. Watson, J. Katakura, K. Okumura, S. Kamada, M. Katoh, and K. Nishimura, "Development of a radiological characterization submersible rov for use at fukushima daiichi," *IEEE Transactions on Nuclear Science*, vol. 65, no. 9, pp. 2565–2572, 2018.

[11] T. Green, K. Kamel, S. Li, C. Shinn, P. Toscano, X. Wang, Y. Ye, and R. Groß, "A minimalist solution to the multi-robot barrier coverage problem," in *Annual Conference Towards Autonomous Robotic Systems*, pp. 349–353, 2021.

[12] S. Patel, F. Abdellatif, M. Alsheikh, H. Trigui, A. Outa, A. Amer, M. Sarraj, A. Al Brahim, Y. Alnumay, A. Felemban, *et al.*, "Multi-robot system for inspection of underwater pipelines in shallow waters," *International Journal of Intelligent Robotics and Applications*, pp. 1–25, 2024.

[13] M. Xanthidis, B. Joshi, J. M. O'Kane, and I. Rekleitis, "Multi-robot exploration of underwater structures," in *IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles CAMS*, vol. 55, pp. 395–400, 2022.

[14] K. Zhang and X. Li, "A graph-based optimization algorithm for fragmented image reassembly," *Graphical Models*, vol. 76, no. 5, pp. 484–495, 2014.

[15] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G^{2}o: A general framework for graph optimization," in *IEEE International Conference on Robotics and Automation*, pp. 3607–3613, 2011.

[16] R. Li, S. Liu, G. Wang, G. Liu, and B. Zeng, "Jigsawgan: Auxiliary learning for solving jigsaw puzzles with generative adversarial networks," *IEEE Transactions on Image Processing*, vol. 31, pp. 513–524, 2021.

[17] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[18] X. Song, J. Jin, C. Yao, S. Wang, J. Ren, and R. Bai, "Siamese-discriminant deep reinforcement learning for solving jigsaw puzzles with large eroded gaps," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, pp. 2303–2311, 2023.

[19] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[20] D. Chicco, "Siamese neural networks: An overview," *Artificial Neural Networks*, pp. 73–94, 2021.

[21] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *European Conference on Computer Vision*, pp. 69–84, 2016.

[22] M. Salehi, A. Eftekhar, N. Sadjadi, M. H. Rohban, and H. R. Rabiee, "Puzzle-AE: Novelty detection in images through solving puzzles," *arXiv preprint arXiv:2008.12959*, 2020.

[23] F. M. Carlucci, A. D'Innocente, S. Bucci, B. Caputo, and T. Tommasi, "Domain generalization by solving jigsaw puzzles," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2229–2238, 2019.

[24] C. Wei, L. Xie, X. Ren, Y. Xia, C. Su, J. Liu, Q. Tian, and A. L. Yuille, "Iterative reorganization with weak spatial constraints: Solving arbitrary jigsaw puzzles for unsupervised representation learning," in *Proceedings of the IEEE/CVF Conference on Ccomputer Vision and Pattern Recognition*, pp. 1910–1919, 2019.

[25] Y. He, B. Lennox, C. Hu, and F. Arvin, "Bubbles-swarm micro surface robots for underwater inspection," in *IEEE International Conference on Mechatronics and Automation*, pp. 305–310, 2024.

[26] E. Ferrante, A. E. Turgut, M. Dorigo, and C. Huepe, "Collective motion dynamics of active solids and active crystals," *New Journal of Physics*, vol. 15, no. 9, p. 095011, 2013.

[27] T. Krajník, M. Nitsche, J. Faigl, T. Duckett, M. Mejail, and L. Přeučil, "External localization system for mobile robotics," in *International Conference on Advanced Robotics*, Nov 2013.

[28] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1251–1258, 2017.

[29] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, pp. 303–338, 2010.

[30] A. Atapour-Abarghouei, S. Akcay, G. P. de La Garanderie, and T. P. Breckon, "Generative adversarial framework for depth filling via wasserstein metric, cosine transform and domain transfer," *Pattern Recognition*, vol. 91, pp. 232–244, 2019.

[31] Y. Liu, Z. Li, H. Liu, and Z. Kan, "Skill transfer learning for autonomous robots and human–robot cooperation: A survey," *Robotics and Autonomous Systems*, vol. 128, p. 103515, 2020.

[32] P. T. Jackson, A. A. Abarghouei, S. Bonner, T. P. Breckon, and B. Obara, "Style augmentation: data augmentation via style randomization.," in *CVPR workshops*, vol. 6, pp. 10–11, 2019.

[33] A. Atapour-Abarghouei and T. P. Breckon, "To complete or to estimate, that is the question: A multi-task approach to depth completion and monocular depth estimation," in *International Conference on 3D Vision*, pp. 183–193, 2019.

[34] X. Liu, C. Yoo, F. Xing, H. Oh, G. El Fakhri, J.-W. Kang, J. Woo, *et al.*, "Deep unsupervised domain adaptation: A review of recent advances and perspectives," *APSIPA Transactions on Signal and Information Processing*, vol. 11, no. 1, 2022.

[35] L. Cheng and S. J. Pan, "Semi-supervised domain adaptation on manifolds," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 12, pp. 2240–2249, 2014.

[36] C. Shen, X. Ji, and C. Miao, "Real-time image stitching with convolutional neural networks," in *IEEE International Conference on Real-time Computing and Robotics*, pp. 192–197, 2019.