Detecting and Suppressing Marine Snow for Underwater Visual SLAM

Lars Martin Hodne[1*] Eirik Leikvoll[1*] Mauhing Yip[2] Andreas Langeland Teigen[1]
Annette Stahl[2] Rudolf Mester[1]

[1]Department of Information and Computer Sciences [2]Department of Engineering Cybernetics
Norwegian University of Science and Technology

Abstract

Conventional SLAM methods which work very well in typical above-water situations, are based on detecting keypoints that are tracked between images, from which egomotion and the 3D structure of the scene are estimated. However, in underwater environments with marine snow — small particles of organic matter which are carried by ocean currents throughout the water column — keypoint detectors are prone to detect the marine snow particles. As the vast majority of SLAM front ends are sensitive against outliers, and the marine snow acts as severe "motion noise", failure of the regular egomotion and 3D structure estimation is expected. For this reason, we investigate the structure and appearance of marine snow and developed two schemes which classify keypoints into "marine snow" or "clean" based on either the image patches obtained from usual keypoint detectors or the descriptors computed from these patches. This way the subsequent SLAM pipeline is protected against 'false' keypoints. We quantitatively evaluate the performance of our marine snow classifier on both real underwater video scenes as well as on simulated underwater footage that contains marine snow. These simulated image sequences have been created by extracting real marine snow elements from real underwater footage, and subsequently overlaying these on "clean" underwater videos. Qualitative evaluation is also done on a night-time road sequence with snowfall to demonstrate applicability in other areas of autonomy. We furthermore evaluate the performance and the effect of marine snow detection & suppression by integrating the snow suppression module in a full SLAM pipeline based on the pySLAM system.

1. Introduction

When applied to underwater scenarios, Visual Odometry and Simultanous Localisation And Mapping (SLAM) face numerous challenges which appear significantly less frequently in regular above-water applications. Such challenges are *e.g.* moving illumination and the reduced zone of usable image landmarks, limited by the illumination cone and the achievable depth of field in a low illumination, turbid environment. Marine snow, the challenge in focus in this paper, describes particles present throughout the water column, ranging from millimeter scale up to decimeter scale [1]. As its name suggests, marine snow can have the appearance of snowfall; its movement is heavily influenced by ocean currents, and under illumination it fills its surroundings with bright white spots. This combination of dynamic motion and an appearance which contrasts with most backgrounds makes the snowflakes salient for keypoint de-

Figure 1. In our approach, we extract marine snow from underwater footage with untextured background (top), and superimpose this snow on arbitrary footage to create labelled training data (bottom images) for training snowflake detectors .

*These authors contributed equally

tectors and constitutes a significant source of motion noise. Thus, keypoint-based SLAM runs a significant risk of producing wrong egomotion estimates or even tracking failure if the number of snowflakes detected, and thus the outlier rate, becomes too high. Of course, this issue also exists for perception in case of heavy snowfall in autonomous driving.

In this paper, we present our approach to mitigate the effect of marine snow by developing two machine-learning systems to filter 'false' keypoints. The main contributions of this paper are:

- We developed two efficient classifiers for marine snow, P-CLAS and D-CLAS; they are designed to run in piggy-back mode on top of arbitrary keypoint detectors — this is done to limit processing to image areas which are actually candidates for being regarded as keypoints. While classifier P-CLAS works on the image area around the detected keypoint, the second classifier, D-CLAS, works on the binary keypoint descriptors provided by the ORB [18] detector/descriptor.
- We investigate how the descriptor-based classifier D-CLAS (which is computationally 'cheaper' than the one working on image patches) compares in performance to the patch classifier P-CLAS. This comparison is done both on a large image dataset as well as for the case of being integrated into a SLAM pipeline.
- We provide a method to extract snow and marine snow from images with essentially untextured backgrounds. We have considerable underwater footage with this characteristic, and thus could collect a huge set of 'ground truth' marine snow examples. The resulting 'snowflake dataset' is used by us for superimposing marine snow on 'clean' images or video sequences. It is publicly available[1], and to our knowledge it is the first of its kind.
- We extend an existing underwater pose-estimation dataset (VAROS [20]) with superimposed marine snow to provide a new benchmark with marine snow motion noise, and accurate ground-truth values.
- We test our method on an above-water snowy sequence, and demonstrate that with further fine-tuning our results should be transferable to the above-water domain.

2. Related Work

Marine snow mitigation for computer vision tasks is a relatively recent research topic. Early methods aimed at a more broad form of image enhancement modelled marine snow as a simple form of additive noise, however, more recent methods aimed specifically at marine snow point out the weaknesses of this approach, like its disregard of properties such as water absorption, size, shape, and backscattering [4].

Most methods pursue marine snow removal in the interest of improving object detection pipelines, and therefore detect snow in the entire image. A family of filter-based approaches for marine snow detection and removal can be traced back to the work of Banerjee *et al*. [2]. It presents a basic approach which does snow removal using median filtering and implicit snow detection based on the luminance channel of a YCbCr (luminance, blue-difference, red-difference) image-representation. The image is traversed with a 7x7 window, and locations which have a high luminance center and high luminance variance are selected for marine snow removal. There exists an extension of this method with multi-scale filters to address particles of varying size, however further details are not given [17].

From Farhadifard *et al*. [7], we find another multi-scale approach, which like earlier filter-based methods uses the dissimilarity of the moving window center value to the window mean as a selection metric. To identify additional outliers within a patch, the patch is represented in RGB colorspace, and an outlier detection step selects all pixels which are closer to the pixel-center than a threshold based on a weighted standard-deviation value. As a final criteria, high-saturation patches are considered false detections, and consequently removed due to the typically grayscale appearance of marine snow.

The paper [6] highlights and addresses one shared shortcoming of the aforementioned methods, namely their implicit dismissal of the temporal information present in video sequences. Allegedly, this is the first spatio-temporal marine snow removal method. From three input frames, the method detects and removes snow in the center frame.

In a 2021 paper, Sato *et al*. [19] state that they are unaware of any deep learning based marine snow removal methods. However, neural networks have been used in an intermediate marine snow detection step before filter-based removal [12]. This method considers the temporal nature of marine snow by utilising 3D neural networks. Their architecture first detects snow using a combination of 3D and 2D convolutions, before using adaptive median filtering to remove the snow.

In another paper [14], the authors do multi-scale detection and removal of above water snow. Their system is divided into three main parts. First, feature maps are calculated at three different scales using a multi-scale Convolutional Neural Network (CNN). Next, the feature maps are concatenated and fed through the snow detection module—a 40-layer modified DenseNet. Finally, to remove the snow, the output from the snow detection module is concatenated with the feature maps from the multi-scale CNN and passed

[1] https://www.ntnu.edu/arosvisiongroup/varos

through yet another densely connected CNN.

With our focus on keypoint classification, performing snow-detection on the full image entails a significant amount of unnecessary computation. A more efficient approach is to only focus on those specific points which are used by the affected SLAM pipeline, *i.e*. the keypoints given by its keypoint detector. Keypoint rejection of stand-alone keypoints is somewhat rare, as most outlier rejection methods are based on a set of matched keypoints from a pair of images. For example, the seminal paper [8] introduces RANSAC, an outlier rejection method which creates motion hypotheses from different subsets of the keypoint correspondences, and tests these estimates against the remaining data.

However, waiting until the matching step is complete wastes resources on matching keypoints which should be removed either way. Therefore, rejecting keypoints as soon as possible, or not detecting them at all should be the preferred method. The workshop paper [9] uses random forest classifiers to predict the suitability of the keypoints for matching, thereby implicitly evaluating keypoints for pose-estimation. Their implementation uses a random forest with 25 decision trees with a maximum tree depth of 25. Importantly, their method classifies on the keypoint descriptors, meaning there is no additional cost related to feature extraction before classification. In scenes with high amounts of foliage or dynamic objects their method removed 70% of the keypoints while retaining 60% of the matches.

The conference paper [10], uses a Support Vector Machine (SVM) to predict the suitability of an image region for image retrieval in geo-localization. To increase the discriminative power of the input, they perform classification based on bundles of descriptors retrieved from the same local image region.

Another paper [13] does keypoint rejection in underwater images for lighting artefacts and dynamic phenomena, such as fishes and caustics, as well as marine snow. They take a 257×257 image patch around each keypoint and scale them down to 65×65. Each patch is classified by a CNN as either suitable or unsuitable for tracking. Their architecture consists of a shallow network with three convolutional layers with ReLU and maxpooling, followed by a fully connected layer and a soft-max layer. Training is supervised, with manually labelled images from other datasets. The proposed real time plug-and-play keypoint rejection system has been verified by comparing drift accumulated by ORB-SLAM [16] and DynaSLAM [3], a SLAM system which accounts for dynamic environments.

3. The ANN-based Approaches to Snow Classification

We created two neural networks for snow classification of keypoints, P-CLAS which extracts image patches from keypoints based on their coordinates such that it can be used with any keypoint-based pipeline, and D-CLAS with descriptors as input, meaning it must be trained for the particular descriptor it is to be combined with. Common to both methods is the Sigmoid activation function in their last layer which makes the final output a pseudo-probability estimate for membership of the positive (snow) class.

Layer	L1	L2	L3	L4	L5	L6	L7
Activation	ReLu						Sigmoid
Dimensionality	256	196	196	128	64	16	1

Table 1. Neural Network architecture for the descriptor classifier D-CLAS

Layer	L1	L2	L3	L4	L5	L6
Layer type	CNN w/ 3x3 filters					Dense
Activation	ReLu					Sigmoid
Input Depth	9	32	64	64	64	256
Input Height/Width	64	32	16	8	4	N/A

Table 2. Neural Network architecture for the image-patch classifier P-CLAS

D-CLAS was designed for ORB-descriptors with a Fully Connected Neural Network (FCNN) architecture (cf. Table 1).

P-CLAS is structured as a multi-scale CNN + FCNN architecture (cf. Table 2). The multi-scale input allows the classifier to discriminate marine snow of different sizes, which is significant because marine snow can vary from a few pixels to 50×50 image regions which can contain multiple undesirable keypoints. With keypoint coordinates as input, we extract patches at three scales ([64×64, 48×48, 32×32]) and, using bilinear interpolation, rescale and subsequently stack them to create a $9 \times 64 \times 64$ input. The network has 5 layers with ReLU, BatchNorm, and maxpooling, and a 6th dense layer with 1 neuron.

Both networks were trained with the Adam optimizer [11]. During training, we frequently validate on the validation splits of the datasets. The models with the highest F2-score in validation were saved for further evaluation.

4. Datasets

We developed our own datasets for training and evaluation. We first collected underwater sequences in which all features are either suitable for SLAM, or all features are marine snow. This means sequences near the ocean floor with no visible marine snow, and sequences distant from both the ocean surface and the ocean floor, in which only marine snow is visible and nothing else. Such sequences circumvent the need to manually label marine snow, which can easily amount to thousands of samples per frame.

With these images, we generate four datasets with full

HD images and keypoints and descriptors labeled as "snow" and "clean". The first Unmodified (U) dataset uses the images as-is to detect keypoints and store their coordinates and descriptors. However, the U dataset has some notable caveats. First and foremost, the presence of marine snow can easily be determined by the colour and texture of the image, since all images of marine snow inevitably come with a background with various shades of blue. Conveniently, we can use these untextured backgrounds to reliably extract marine snow and superimposing it onto more varied backgrounds, using a weighted sum (alpha-keying). With the extracted snow, we create three datasets named Underwater (UW), Overwater (OW), and Snowy-VAROS. These are discussed later in this section.

To extract snow, we use a strided window approach with stride 10. In each 60×60 window, P, we calculate the Euclidean RGB-distance, D, of each individual pixel value at location $p \in P$ to the median colour M_P of the window. As indicated in Eqs. (1) and (2), these distances are scaled by the inverse maximum distance to create a pixel-wise weighting between 0 and 1. The weight, W_p, is set to 0 if $D(M_P, p)$ is below a threshold value $\tau_D = 30$, or if the grayscale intensity of p, $I_{GS}(p)$, is below $\tau_I = 20$.

$$W_p = \begin{cases} 0 & \text{if } I_{GS}(p) < \tau_I \\ 0 & \text{if } D(M_P, p) < \tau_D \\ \dfrac{D(M_P, p)}{\max\limits_q D(M_P, q)} & \text{otherwise.} \end{cases} \quad (1)$$

where

$$D(p, q) = |I(p) - I(q)|. \quad (2)$$

For each pixel, the average weight across all windows is used when extracting snow. This is to ensure that windows with large marine snow particles which shift the median colour away from the background colour do not introduce unwanted artefacts when superimposing the snow. With a background image B, alpha-key weight W, and snowy image S, we superimpose images according to Equation 3:

$$I = B \odot (1 - W) + S \odot W. \quad (3)$$

After extracting snow from the snowy images in the U dataset, we create the three other datasets. The Underwater (UW) dataset superimposes the extracted snow onto the remaining images in U which are free from snow. The Overwater (OW) dataset uses above-water images from the *Exclusively Dark Images Dataset* (ExDark) [15] as backgrounds for superimposing to introduce more variation in the background images. Images in the ExDark dataset which featured rain, starry skies, or snowfall were removed because of their exceptional similarities to marine snow.

Finally, to demonstrate the flexibility of our superimposing approach we use it to add a video sequence of snow

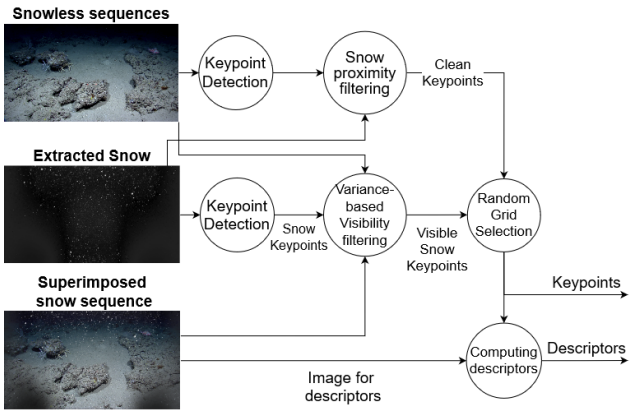

Figure 2. Data generation pipeline with superimposed snowy sequences

Figure 3. Snow superimposed on the VAROS dataset

to the synthetic underwater benchmarking dataset VAROS [20]. This has the benefit of a known camera matrix and ground-truth pose. By superimposing snow from a video sequence, we ensure that the motion of the snow is consistent between frames. A sample frame from this Snowy-VAROS sequence is seen in Figure 3.

To create keypoint coordinates and descriptors for training, we perform keypoint detection separately on the background image, and extracted snow image, as opposed to detecting keypoints on the combined, superimposed image. This was done primarily to increase the number of keypoints on marine snow, and because it makes for a more difficult dataset in which some of the detected snow is less visible than normal. Figure 2 presents the pipeline used to generate keypoints for the datasets from a three-tuple of extracted snow, background, and their combined image. A challenge of this approach is that snow can be superimposed either over a good keypoint in the background image or in an image region where the snow is not visible, meaning keypoints can become mis-labeled in the combined image. Consequently, a keypoint, K_s, detected on the extracted snow is rejected if inequality 4 is not true, where

	Images	Snow KPs	Background KPs
Unmodified	6,008	598,931	1,181,570
Underwater	10,051	1,525,556	2,227,102
Overwater	8,705	1,772,123	2,055,001
Total	24,764	3,896,610	5,463,673

Table 3. Datasets and their sizes. Train, val and test splits were made following the 80/10/10 convention

P_{Si}, and P_{BG} are image patches of K_s in the superimposed image and background image, respectively, and $\mathcal{E} = 14$ is an empirically selected threshold.

$$\mathsf{Var}\left[P_{Si}\right] > \mathsf{Var}\left[P_{BG}\right] + \mathcal{E} \tag{4}$$

Secondly, to verify that keypoints detected on the background image can not be perceived as mis-labelled after superimposing due to abutting snow, we verify that the maximum color channel value of a small 8×8 neighbourhood surrounding this keypoint within the extracted snow is below an empirically selected threshold, $\tau_S = 70$. Finally, we divide the image into a 10×10 grid and select keypoints at random from these bins to limit the dataset size, and to reduce the presence of overlapping samples. Importantly, the ORB descriptors are still generated on the combined image.

5. Experiments

We conducted experiments to evaluate stand-alone classification performance, and performance in SLAM use cases. For stand-alone performance, we used test splits of our U, OW, and UW datasets and evaluated F1 score, accuracy, True Positive Rate (TPR) and True Negative Rate (TNR). The datasets and their sizes are listed in Table 3.

Qualitative assessments of keypoint classification were performed on four diverse underwater sequences, each pictured in Figure 4, by extracting 2000 keypoints with the ORB detector and classifying these frame-by-frame.

For evaluation in SLAM use-cases, we implemented our classifiers into the pySLAM framework[2] which offers a very customiseable SLAM-platform intended for experimentation and education. pySLAM features most of the expected attributes of a modern SLAM-system, including keyframe management, local and global bundle adjustment, outlier rejection with RANSAC, ratio testing, and motion models with active matching [5]. Our experiments in pySLAM were done on the synthetic VAROS and Snowy-VAROS sequences.

5.1. Binary classification metrics

While training classifiers, we store the checkpoint which achieved the best F2 score on the validation data-split. In Table 4, we list these models, and their binary classification metrics on the separate test-splits of our datasets.

It is clear that both D-CLAS and P-CLAS have learned the classification task successfully, yet P-CLAS maintains remarkable results on most datasets, outperforming the descriptor classifier in all datasets. However, D-CLAS has an unavoidable benefit in that it requires no pre-processing of the image if descriptors are present, and can operate far more efficiently, surpassing speeds of 66000 keypoints per second, compared to 14600 for P-CLAS, both on a GTX 1080 GPU. However, our testing shows that these differences can be explained by overhead from patch-extraction, which can be improved compared to our implementation since it assumes that keypoints in the same batch come from different images, which is true for training, but otherwise is typically false.

Both classifiers, when trained on the U dataset, score high on the U test-split. However, we notice a decrease in TPR when the superimposed OW and UW datasets are included in the test data. This strongly suggests that U-trained models only learn to recognise white blobs on an untextured background, hence when more textured backgrounds appear, the number of false negatives increase. P-CLAS in particular, seems to rely too much on the predictable backgrounds of the U training data, since its TNR is high on both the U testset, and the unmodified + UW testset. This suggests that training on varied backgrounds, and thus the superimposed datsets, is particularly important for P-CLAS models, since they will otherwise latch onto background characteristics which are not encoded by the descriptors. To be clear, the near perfect scores on U, highlight the simplicity of the U datasets, rather than the prowess of the methods.

On the topic of what the descriptor encodes, it is feasible that the discrepancy which is consistently present in all testsets between P-CLAS and D-CLAS can be explained by the CNN being able to use more contextual clues from the background which are not available from descriptors. This could lead to P-CLAS models performing better than D-CLAS models when encountering backgrounds familiar from training, but worse on unfamiliar backgrounds.

When it comes to networks trained on superimposed data, the models which were trained on all datasets performed the best on every testset, except the U testset. Even if the test dataset only included two of the three datasets used in training, training on every dataset gave the best overall performance which could indicate an improved ability to generalise to unseen data.

5.2. Qualitative results in Keypoint Classification

We begin with video A) in Figure 4, which features a smooth ocean floor with small mounds of sand, and a somewhat dense cover of small and bright marine snow. Some spots on the ground can be mistaken for marine snow in still

[2] https://github.com/luigifreda/pyslam

		Unmodified				UW + U				OW + U				All datasets			
		F1	Acc	TPR	TNR	F1	Acc	TPR	TNR	F1	Acc	TPR	TNR	F1	Acc	TPR	TNR
Patch	U	**0.999**	**0.999**	**0.999**	**1.0**	0.617	0.792	0.446	0.999	0.47	0.687	0.309	**0.997**	0.402	0.692	0.252	**0.998**
	UW + U	0.968	0.972	0.991	0.957	0.948	0.961	0.931	0.98	0.854	0.861	0.904	0.825	0.877	0.897	0.894	0.9
	OW + U	0.998	0.998	0.996	0.999	0.913	0.94	0.84	**0.999**	0.94	0.948	0.894	0.993	0.91	0.932	0.839	0.996
	All	0.996	0.996	0.996	0.997	**0.975**	**0.982**	**0.955**	0.998	**0.961**	**0.965**	**0.954**	0.975	**0.964**	**0.971**	**0.95**	0.985
Desc	U	0.945	0.951	0.98	0.929	0.778	0.848	0.712	0.929	0.809	0.833	0.784	0.873	0.763	0.82	0.707	0.899
	UW + U	0.944	0.949	0.978	0.927	0.913	0.933	0.931	0.935	0.892	0.898	0.943	0.86	0.893	0.909	0.93	0.895
	OW + U	0.954	0.959	0.977	0.946	0.916	0.937	0.917	0.949	0.917	0.925	0.919	0.93	0.909	0.926	0.906	0.939
	All	0.955	0.961	0.964	0.958	0.935	0.952	0.926	0.967	0.919	0.928	0.912	0.941	0.921	0.936	0.909	0.955

Table 4. Binary classification results by the classifiers. Rows denote the the network and its training data, while columns denote the test dataset. The Unmodified dataset (U), Underwater superimposed (UW), and Overwater superimposed (OW) were combined and used for testing. We provide F1-scores, accuracy, True Positive Rates (TPR), and True Negative Rates (TNR) for each case.

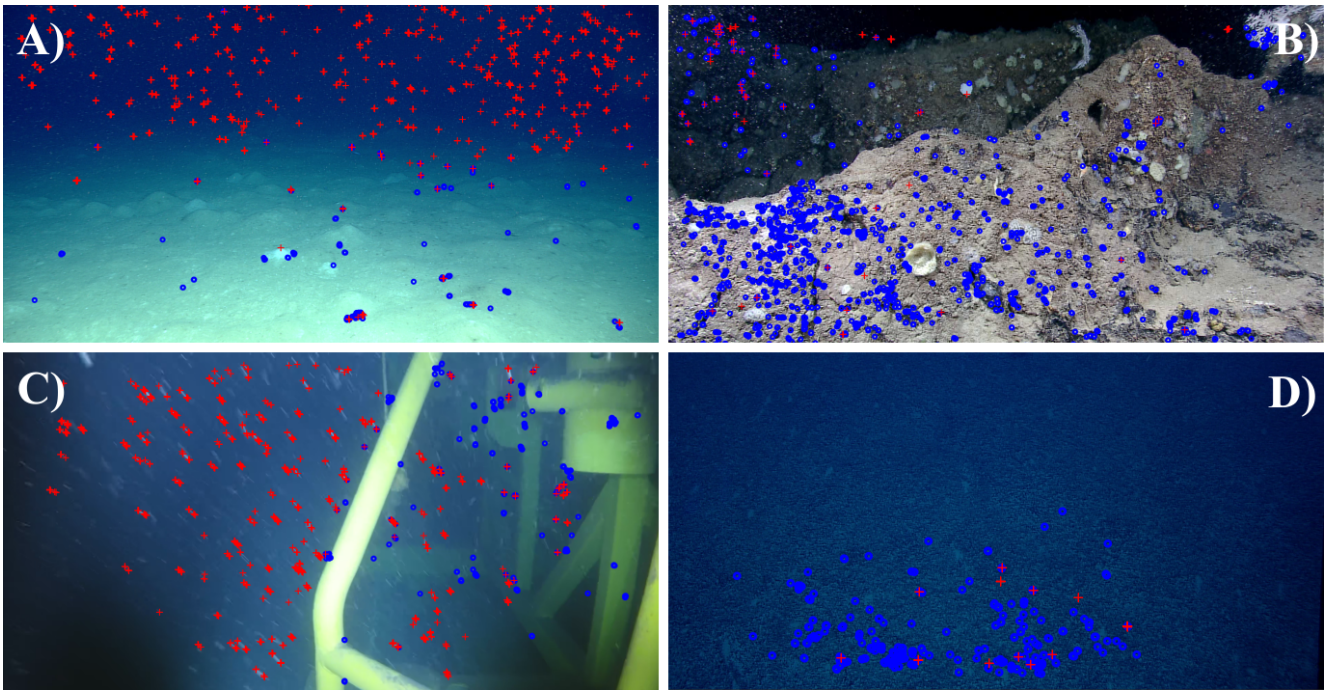

Figure 4. Frames from the four videos used for visualisation of keypoint classification. Red crosses indicate a snow classification, while blue circles indicate a clean keypoint classification. 2000 Keypoints were detected with ORB per frame.

images. Models tested on this video mainly struggled with classifying marine snow when transitioning between the textureless region in the upper half of the image, and the textured region below. With P-CLAS models, those trained on the U dataset and at least one superimposed dataset seemed to handle this issue the best. This was true also for D-CLAS models, though these showed slightly worse False Negative Rates, and False Positive Rates on the ground.

Video B), features a complex structure of large, jagged, and overlapping rocks. Of particular interest is the marine snow seen in the top left corner during the beginning of the sequence, which is strikingly bright and visible, despite the rocks in the background. Thus, this sequence offers a break from the typically textureless backgrounds which are far more common. Performance on this sequence was particularly bad from P-CLAS models trained only on the U-data, which labelled all keypoints as "clean". Other P-CLAS models trained on superimposed data were able to improve upon this, but none were able to compete with D-CLAS models, not even U-trained D-CLAS models. D-CLAS models were hard to tell apart, but it seems like the OW-trained model did worse, and both the U-trained model and OW-trained model did best. These results may be an indication that P-CLAS models rely more on properties of the background when classifying. False Positive Rates were very low for all models on this sequence.

Video C) features a bright yellow charging station, with extreme amounts of large marine snow particles. This sequence highlights a weakness of current feature detectors underwater, in the sense that the sequence's lack of corners

and blobs (other than marine snow) leaves most corner detectors with nearly no useful features, *e.g.* for pose estimation. Despite a limited presence of useful keypoints, the classifiers labelled many keypoints as clean. While TPR rates were good on this sequence, false negatives were also seen frequently. In the final part of this sequence, the robot moves rapidly, giving the marine snow a stretched appearance which is not present in the training data, leading to high amounts of false negatives. While all classifiers exhibited this trait, D-CLAS models were the worst afflicted.

Notably, models which were not trained exclusively on the OW dataset showed a tendency to switch from a true positive detection to a false negative when snow particles moved in front of an uncommon background, e.g., the yellow beams of the charger sequence. However, OW-trained models struggled more with classifying snow in textureless regions. All classifiers struggled with the sequence's largest snow particles which are particularly close to the camera.

As a final test on False Positive Rates, video D) deliberately features an insignificant amount of snow, yet in certain frames, the pebbled texture of the ocean floor carries an appearance somewhat (though not completely) reminiscent of marine snow. Most classifiers tested on this sequence, be it patch or descriptor based, were not prone to mislabel the ground as snow, with P-CLAS models generally achieving near-perfect accuracy, and D-CLAS models not far behind. However, the P-CLAS model which was trained on both the underwater superimposed data and unmodified data was a curious exception. Once the keypoint detector began detecting on the ground, most keypoints were classified incorrectly as snow. This continued as the camera came closer and the number of ground keypoints increased, but eventually stopped once the robot came even closer to the ground and the likeness to marine snow disappeared.

To summarise these results, P-CLAS models typically perform better than D-CLAS ones if the background is known from training. With textured backgrounds, performance drops off, in which case training with superimposed datasets can help, but not completely. Compared to the image-patches, ORB-descriptors seem to encode less information about the background, which can remove irrelevant information, and in certain instances help classification, such as in video B) where D-CLAS models consistently outperformed P-CLAS models. However, it could be the case that too much information is lost through the ORB representation, such that overall performance is reduced.

5.3. Qualitative results in an above-water sequence

To examine generalisability and applicability on a broader range of tasks, we visualised classification on a night-time road sequence with real snowfall, using an OW+U-trained P-CLAS model. The sequence features a twisting, snow-covered road with dimly lit trees on both sides and snowfall illuminated by the car's headlights. At the bottom of the image is the contour of the car's dashboard. Performance on this sequence was mixed, but showed some promise. Clean points placed on the roadside, dashboard, and trees are typically classified correctly. The same goes for snow keypoints in the darker regions of the image. However, one struggle of the classifiers is the snow just in front of the right headlight which is particularly bright in front of a white background. This kind of image patch is not found in the training data, so unsurprisingly it is classified incorrectly. However, considering the overall performance on this sequence it seems probable that given finetuning on above-water data, our results should be transferable to the road domain as well. Generally, in above-water scenarios snow often appears on more textured backgrounds, which can be a source of decreased performance not covered by this particular video. On the other hand, increased illumination during the daytime can make the snow less prominent in some footage.

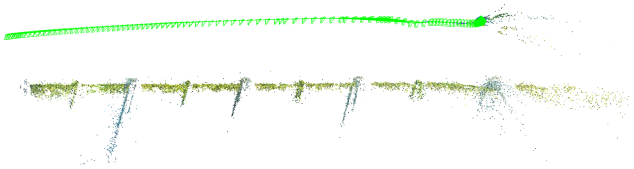

Figure 5. A point map from running pySLAM on VAROS without snow and no classification. The pipe is properly tracked.

5.4. Qualitative results with pySLAM

Testing SLAM performance on real-world sequences has the potential to give the most realistic view of the effect of snow classification. However, by using the synthetic VAROS sequence with and without superimposed snow, we are able to control the difficulty of both the background sequence and snow conditions. Furthermore, we are able to compare results between Snowy-VAROS and the original, snow-free VAROS sequence which lets us evaluate the results of keypoint rejection more definitively than most qualitative tests. However, we must expect that models trained on superimposed images perform disproportionally better on Snowy-VAROS, due to similarities in the superimposing process of Snowy-VAROS and the training datasets. We choose a subsequence of VAROS in which the robot travels adjacent to a straight pipe (see Fig. 3). This pipe offers more defined features for keypoint detection compared to other sections of VAROS, and makes it easy to judge the tracking quality by how accurately the straight pipe is mapped.

When testing with pySLAM alone on the Snowy-VAROS sequence, a considerable amount of keypoints are detected on snow, which lead to rapid tracking failure and inconsistent behaviour between runs. During some runs,

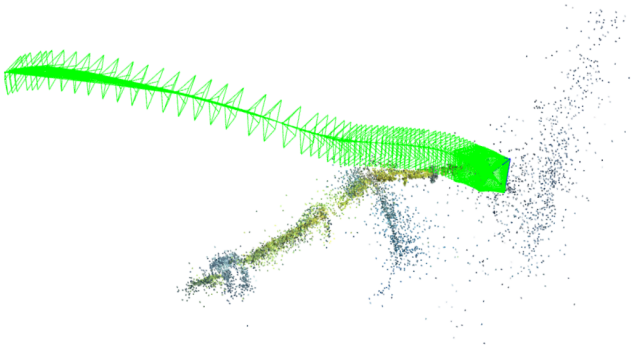

Figure 6. A point map from running pySLAM without keypoint classification on Snowy-VAROS. The path stops in a wall of snow and the pipe appears to bend, unlike the source video.

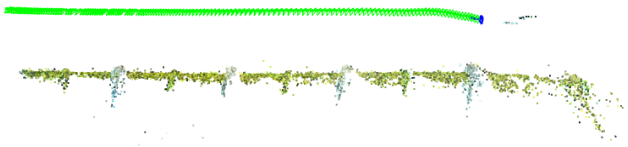

Figure 7. A point map from running pySLAM with keypoint classification on Snowy-VAROS. The pipe is properly tracked, and few snow keypoints are added to the map.

tracking fails completely, while in other runs the tracking is closer to the movement seen in the sequence. When visualising the sparse map made by pySLAM, seen in Figure 6, we see what looks like a wall of snow prominently in the map. Furthermore, the pipe which is completely straight in the sequence appears bent in the point cloud.

Both P-CLAS and D-CLAS stabilised tracking in pySLAM, to the extent that they were difficult to tell apart. While they were unable to remove all unreliable points, pySLAM continued tracking for far longer and was far more reliable, giving consistent tracking outputs between runs. An example can be seen in Figure 7, where the pipe appears straight in the point cloud like it should, with the exception of the very end. This behaviour is similar to that seen in the VAROS sequence without snow and no classification, as seen in Figure 5, and occurs because pySLAM is unable to detect a sufficient amount of good keypoints, irrespective of the presence of snow. Since P-CLAS and D-CLAS differed in earlier testing, their comparable performance with pySLAM could indicate that as long as the number of marine snow keypoints is reduced such that the snow is no longer dominating the RANSAC motion hypotheses, tracking can continue with traditional outlier rejection. On Snowy-VAROS, out of 3000 features, D-CLAS removed 1,627 keypoints and P-CLAS removed 1,366 keypoints in each frame on average.

For comparison, we run pySLAM on the original VAROS dataset, which has nothing resembling marine snow. With snow rejection enabled on the unmodified VAROS sequence, out of 3,000 keypoints, we see on average 36 and 201 rejections, *i.e.*, false positives, by D-CLAS and P-CLAS, respectively.

6. Conclusion

In this paper we have demonstrated two methods for classification of keypoints obtained from the ORB detector [18] in order to suppress the effect of marine snow. The methods can be used to aid pose estimation, create keypoint detection masks or assist in underwater image restoration. Our results show that classifying snow, either with ORB descriptors or image patches, can achieve near perfect performance for snow in front of an untextured background. To enable snow detection also on textured backgrounds, additional training data is necessary. We created such data by extracting snow from underwater footage with untextured background. This allowed us to overlay real marine snow on arbitrary image material. Despite a lack of training on such scenes, initial experiments on a night-time driving sequence featuring snowfall suggest that the classifiers can be applied in above-water scenarios with some further finetuning. Using the pySLAM framework we demonstrated how our method can be incorporated as a keypoint rejection component in a SLAM pipeline. We showed that our methods were able to overcome the difficulties that a SLAM system with standard outlier removal has with underwater footage affected by marine snow. We provide the snow dataset to the public in order to foster further research on the challenging topic of underwater and above-water SLAM under difficult visibility conditions. Extensions of our research could examine other descriptors than ORB, novel classifiers, and new methods of extracting snow.

References

[1] Alice L. Alldredge and Mary W. Silver. Characteristics, dynamics and significance of marine snow. *Progress in Oceanography*, 20(1):41–82, 1988. 1

[2] Soma Banerjee, Gautam Sanyal, Shatadal Ghosh, Ranjit Ray, and Sankar Nath Shome. Elimination of marine snow effect from underwater image - an adaptive probabilistic approach. In *2014 IEEE Students' Conference on Electrical, Electronics and Computer Science*, pages 1–4, 2014. 2

[3] Berta Bescos, José M. Fácil, Javier Civera, and José Neira. Dynaslam: Tracking, mapping, and inpainting in dynamic scenes. *IEEE Robotics and Automation Letters*, 3(4):4076–4083, 2018. 3

[4] Matthieu Boffety and Frédéric Galland. Phenomenological marine snow model for optical underwater image simulation:

This paper is financially supported by the Norwegian Research Council in the project Autonomous Robots for Ocean Sustainability (AROS), project number 304667.

Applications to color restoration. In *2012 Oceans - Yeosu*, pages 1–6, May 2012. 2

[5] Margarita Chli and Andrew Davison. Active matching. pages 72–85, 10 2008. 5

[6] Boguslaw Cyganek and Karol Gongola. Real-time marine snow noise removal from underwater video sequences. *Journal of Electronic Imaging*, 27:1, 07 2018. 2

[7] Fahimeh Farhadifard, Martin Radolko, and Uwe von Lukas. Single image marine snow removal based on a supervised median filtering scheme. In *VISIGRAPP*, 2017. 2

[8] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, jun 1981. 3

[9] Wilfried Hartmann, Michal Havlena, and Konrad Schindler. Predicting matchability. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6 2014. 3

[10] Hyo Jin Kim, Enrique Dunn, and Jan-Michael Frahm. Predicting good features for image geo-localization using per-bundle vlad. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015. 3

[11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 3

[12] Michał Koziarski and Bogusław Cyganek. Marine snow removal using a fully convolutional 3d neural network combined with an adaptive median filter. In Zhaoxiang Zhang, David Suter, Yingli Tian, Alexandra Branzan Albu, Nicolas Sidère, and Hugo Jair Escalante, editors, *Pattern Recognition and Information Forensics*, pages 16–25, Cham, 2019. Springer International Publishing. 2

[13] Marco Leonardi, Luca Fiori, and Annette Stahl. Deep learning based keypoint rejection system for underwater visual ego-motion estimation. *IFAC-PapersOnLine*, 53(2):9471–9477, 2020. 21th IFAC World Congress. 3

[14] Pengyue Li, Mengshen Yun, Jiandong Tian, Yandong Tang, Guolin Wang, and Chengdong Wu. Stacked dense networks for single-image snow removal. *Neurocomputing*, 367:152–163, 2019. 2

[15] Yuen Peng Loh and Chee Seng Chan. Getting to know low-light images with the exclusively dark dataset. *Computer Vision and Image Understanding*, 178:30–42, 2019. 4

[16] Raúl Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015. 3

[17] Martin Radolko, Fahimeh Farhadifard, and Uwe Freiherr von Lukas. Dataset on underwater change detection. In *OCEANS 2016 MTS/IEEE Monterey*, pages 1–8, 9 2016. 2

[18] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International Conference on Computer Vision*, pages 2564–2571, 2011. 2, 8

[19] Yuya Sato, Takumi Ueda, and Yuichi Tanaka. Marine snow removal benchmarking dataset, 2021. 2

[20] Peder Georg Olofsson Zwilgmeyer, Mauhing Yip, Andreas Langeland Teigen, Rudolf Mester, and Annette Stahl. The varos synthetic underwater data set: Towards realistic multi-sensor underwater data with ground truth. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 3722–3730, 10 2021. 2, 4