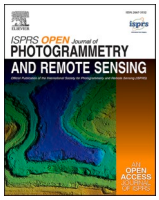
ISPRS Open Journal of Photogrammetry and Remote Sensing 13 (2024) 100072

Contents lists available at ScienceDirect

ISPRS Open Journal of Photogrammetry and Remote Sensing

journal homepage: www.journals.elsevier.com/isprs-open-journal-of-photogrammetry-and-remote-sensing



Robust marker detection and identification using deep learning in underwater images for close range photogrammetry

Jost Wittmann [a,*], Sangam Chatterjee [a], Thomas Sure [b]

[a] Justus Liebig University Giessen, Institute of Experimental Physics I and Center for Materials Research (LaMa), Heinrich-Buff-Ring, 1635392, Gießen, Germany
[b] University of Applied Sciences Mittelhessen, IOM - Institute for Optics and Microsystems, Wiesenstraße 14D, 35390, Gießen, Germany

ARTICLE INFO

Keywords:
Underwater
Photogrammetry
Marker
Machine learning

ABSTRACT

The progressing industrialization of oceans mandates reliable, accurate and automatable subsea survey methods. Close-range photogrammetry is a promising discipline, which is frequently applied by archaeologists, fish-farmers, and the offshore energy industry. This paper presents a robust approach for the reliable detection and identification of photogrammetric markers in subsea images. The proposed method is robust to severe image degradation, which is frequently observed in underwater images due to turbidity, light absorption, and optical aberrations. This is the first step towards a highly automated work-flow for single-camera underwater photogrammetry. The newly developed approach comprises several machine learning models, which are trained by 10,122 real-world subsea images, showing a total of 338,301 photogrammetric markers. The performance is evaluated using an object detection metrics, and through a comparison with the commercially available software Metashape by Agisoft. Metashape delivers satisfactory results when the image quality is good. In images with strong noise, haze or little light, only the novel approach retrieves sufficient information for a high degree of automation of the subsequent bundle adjustment. While the need for offshore personnel and the time-to-results decreases, the robustness of the survey increases.

1. Introduction

Optical surveying methods represent an active and evolving field of research with broad applications. Their relevance in subsea environments extends across various industries, offering a swift and precise means of acquiring 3D data that is easily interpretable. Attempts to construct photogrammetric models of underwater scenes based on natural features have encountered significant challenges, including issues related to backscatter and suboptimal lighting conditions. These challenges often result in incorrect correspondence matching between neighboring images. This is why the authors focus on a purely marker-based approach, and rejects tie points and dense image matching in this work. The strategic deployment of markers enhances reliability and accuracy, making marker-based photogrammetry better suited for the intricacies of the underwater environment. For instance, marker-based photogrammetry demonstrates exceptional accuracy, maintaining a high precision over distances of up to 100 m. A common task in the offshore energy industry is “spool piece metrology,” involving the acquisition of precise 3D distances and angles between distant flanges to design and manufacture connecting pipes. Accurate survey results ensure a seamless installation process, allowing the rigid pipe to connect flanges without mechanical stress, with the majority of the weight resting on the sea floor.

Some of the unique advantages of marker-based photogrammetry are.

- It surpasses the range-limitations of other optical survey methods and tools such as 3D-scanners. Most 3D-scanners are mounted on tripods and get deployed on the sea floor. They utilize the time-of-flight principle. A vertically rotating laser emitter and time-of-flight capable sensor are mounted on a vertical axis, in order to cover 360 deg. field of view. These sensors come with two major limiting aspects: firstly, the laser light is entirely absorbed by the water after about 40 m. This limits the measurement distance to about 20 m, since the return way of the light has to be considered. Secondly, the rotation of the laser sensor around the vertical axes is performed by an under water motorized stepper motor. Under water stepper motors come with a decreased accuracy in comparison to the

* Corresponding author.
E-mail addresses: Jost.Wittmann@SubseaScanning.com (J. Wittmann), Sangam.Chatterjee@physik-uni-giessen.de (S. Chatterjee), Thomas.Sure@me.thm.de (T. Sure).

https://doi.org/10.1016/j.ophoto.2024.100072
Received 15 January 2024; Received in revised form 12 July 2024; Accepted 12 July 2024
Available online 14 August 2024

rest of the sensor system. A common workaround is the deployment of 3 dimensional targets to create a network of reference points, and to re-position the scanner within that network. This process allows to compute the new position of the sensor, and to record new 3D-points with the goal to extend the initial measurement range. Besides the accumulating inaccuracies during the procedure, the resolution of the 3D-points decreases with the measurement distance, making it challenging to maintain a competitive accuracy over larger distances.

- It delivers a high accuracy for distance measurements over 100 m and beyond. This has been verified by the author and his surrounding team, utilizing multiple, calibrated and temperature corrected scalebars with a length of 30 m each.
- It is a proven and highly reliable technique.
- Deployed scale-bars lead to an increased accuracy of the model as well as enabling testing the length measurement error.
- The markers provide high redundancy, which paths the way for a high degree of automation. It has also the potential to lead to high robustness of the bundle adjustment and the camera calibration.

We aim to increase the efficiency of today's workflow: Today's commercial photogrammetry software is not optimized for underwater images, as their detectors for fiducial markers require good image quality in order to work properly. See Fig. 1b. When faced with lower image quality such as shown in Fig. 1c and d, data specialists are compelled to engage manually. This becomes especially challenging in scenarios like the spool piece metrology, where prompt delivery of the results is demanded. Since transferring images from a ship to the shore is not possible due to satellite data transfer limitations, the data specialists are to be present on board vessel - rendering today's workflow inefficiency and expensive. Introducing Machine Learning (ML) for detecting and identifying photogrammetric markers in low contrast images, represents a significant step to overcome the mentioned challenges. Our approach involves performing all image measurements automatically, storing the extracted information in coordinate-files. Having a very small file size, the coordinate files may easily get transferred to shore. To ensure a successful bundle-adjustment, we establish a threshold requiring the correct detection and identification of at least six coded targets per image. This ensures robust performance in diverse conditions. Additional helpful context is either already present due to the previously conducted project planning, or may get communicated by phone or email.

Summarizing the authors contributions with this work.

- High detection and identification rate of targets in versatile underwater visibility. Minimum of six targets per image.
- Fully automated image measurements.
- Small file-size for resulting coordinate files.

2. Related works

We relate to several approaches for the detection of markers. The following two sections differentiate between non-ML and ML approaches. A selection of exemplary works are listed.

2.1. Non-ML-based approaches

Non-ML-based approaches are often based on a sequence of image processing algorithms, many being available to the community through toolkits and libraries. While being well documented and reliable, more complex algorithms require an adjustment of their parameters in dependence of the image to process, as well as long processing time.

ARToolkitX (ARToolworks and Artoolkitx) consists of a collection of software tools, with publicly available source code. Its intended field of application is augmented reality, including geometric and photometric registration. Square targets are detected by calculating the correlation coefficient between the standardized image and a standardized reference database.

The team around Raquel Dosil (Dosil et al., 2013) proposes to detect fiducial markers using an algorithm that reveals radial symmetry. The radial symmetry is computed as the product of the responses of a set of even symmetric feature detectors, with different orientations. The decoding of the binary code is performed using segmentation and conventional decoding.

Wang and Olsen (Wang and Olson, 2016) introduce an enhanced detector for AprilTag2. The process starts with a binarization of the images applying adaptive thresholding. Afterwards, a segmentation into connected components is performed using the union-find algorithm. Finally, quadrats are fit to each cluster of unordered boundary points, partitioning the points into four groups corresponding to line segments. From the line-segments, the markers are reproduced.

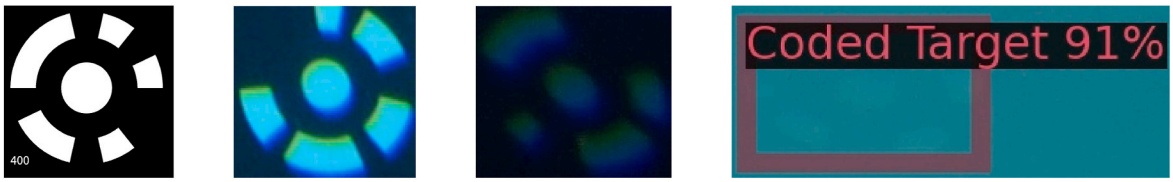
Drap et al. (2013) deploy APRILTags in a rectangular setup, surrounding the corals of interest. The targets are detected by binarizing the image, and then extracting contours. The center of the marker is computed by calculating the weighted average of the gray-scale neighbors.

Cejka et al. (2019) present an enhancement of the ARUco marker detector for square markers. The team applies image processing methods like thresholding, masking areas that do not contain markers and filtering. Finally, contours are detected and the markers get identified. The focus of the team also lies on generating synthetic images, glowing markers and turbid underwater environments.

2.2. ML-based approaches

ML-based approaches offer the advantage of adaptability, allowing them to apply learned knowledge to new scenarios without the need for parametrization during processing. Moreover, modern ML frameworks typically require minimal processing time.

Zhang et al. (2022) propose DeepTag, a framework that supports the detection of a variety of existing marker families. The team suggests a square marker design as well as a detection algorithm using convolutional neural network. The network detects marker ROI's in a top-down manner by directly determining their existence along with non-colinear



(a) Technical drawing.

(b) Good image quality.

(c) Medium image quality

(d) Detected marker with poor image quality. Our method predicts a 91 % probability for the classification as "coded target".

Fig. 1. Examples of the image quality observed underwater for photogrammetric markers, Schneider (Schneider, 1991). In the images b) and c), chromatic aberration is present - a frequent issue when using a flat port. In image d), only little signal is present. Yet, our method successfully detects the marker.

key-points. The authors create artificial training data by adding various noise and (motion-) blurs, and focus on the detection robustness comparing different types of markers.

Jiang et al. (2021) pre-train deep learning network with 'You only look once' (YOLO), a state-of-the-art, real-time object detection system. Jiang uses a data set of 2480 pictures taken in air comprising targets which follow the same design as the ones used in this work, but for the binary code bit, which comprises 12 bits (instead of 20). The pre-trained model is then fine-tuned using 1892 pictures taken underwater. After the target marker is detected, traditional image processing methods are used to recognize the marker. The performance is evaluated comparing the results from the model with traditional image processing methods.

Many of the addressed researchers prefer 2D-Barcodes because square markers encode more symbols and are more robust to noise. In contrast, we focus on circular targets from Schneider as shown in Fig. 1a. Their center may get located more precisely (Drap et al., 2013), if the ellipse eccentricity is considered. Also, they come with less details, enabling their detection at large distances. Schneider detectors are only available in propriety software.

Based on experience, solutions developed for in-air purposes only perform well under water if the visibility is good - and fails in degraded visibility. In contrast, approaches focusing on under water application promise an improved performance. But they strongly depend on a large variety of real-world data during development and testing. Similar challenges apply to solutions based on ML: ML comprises the capability to abstract what it has learned, and consequently performing well in varying conditions. Since acquiring sufficient training data under water is very costly, artificially generated data is often used instead. This again bears the risk that the resulting model only performs well on artificial data, and fails to perform well in real world scenarios. In summary, it can be said that a solution comprising ML trained on a sufficient amount of versatile, real world underwater images has good chances of enabling a highly automated photogrammetric workflow.

3. Methodology

We divide our work-flow into three segments, each contributing to the critical tasks of the marker detection and marker identification. Each segment is implemented into a software-module, which are sequentially executed. This modular design allows us to easily test and compare alternative approaches. Machine-learning is applied to maximize the reliability in each of the three modules. All ML-models were trained using supervised training, and are supplemented by conventional methods. This comprises image-processing such as contrast enhancement, thresholding and edge detection, as well as geometric operations such as coordinate-transformations and geometry fittings. Fig. 2 provides an overview.

The images used for training of the ML-models were acquired between the years 2000 and 2018. The majority was recorded on the Norwegian Continental Shelf by work-class remote operated vehicles (ROV). Image acquisition took often place in several hundred meters depth and at temperatures between −2 and 10 deg.[1] The cameras have undergone upgrades over the years, with the majority of images captured using Nikon D300 and D300S - both single-lens reflex cameras (DSLR), equipped with the 16 mm lens from Carl-Zeiss "Distagon". In recent times, the mirrorless Sony A7 has taken over for the DSLRs. The color-images are recorded in RAW-format, and converted into jpg before further processing.[2] The image database comprises more than 20,000 images with a resolution between 4288 pixels by 2848 pixels and 4912 pixels by 7360 pixels.

This large collection of marker-rich images have been taken in a wide range of visibility conditions. This dataset significantly contributes to our goal of building a reliable and real-world solution. Only marker of the type "Schneider" (see Fig. 1a) as well as non-coded circular markers have been used.

We will now present an overview of our approach. Technical details regarding the implementation and the applied ML-frameworks will be provided in the following chapter 4 "Software Design".

As an initial step in our processing chain, we verify if preprocessing the images leads to an improved marker detection rate. We compare and evaluate four different preprocessing methods: 1. Original images: no preprocessing is applied. This model serves as a reference to evaluate the other three methods. 2. Contrast Limited Adaptive Histogram Equalization: we apply CLAHE to enhance the contrast, specially in contrast poor regions of the images. Such regions appear for example due to strong camera angles, leading to different distances between the camera/light and the object. The effect is reinforced by the attenuation of the light by the water. 3. Kirsch filter: we apply edge-detection using the Kirsch-operator in order to enhance the edges of the markers. Empirical tests have shown that the Kirsch filter enhances edges better than other detectors in underwater imagery. It is well suited for radial objects. 4. The "3-steps method": we apply a processing chain comprising three preprocessing methods: a) CLAHE, b) Bilateral filter, c) Kirsch filter. The bilateral filter serves the suppression of background noise, leading to a reduced amount of false positive detections. Background noise occurs frequently in underwater images due to rock dump on the seafloor. The required parameters for the 3-step-method are determined by a neural network. The network receives the original images as input, and returns suitable parameters in return. Then, the actual image processing is performed. Table 1 lists the 3 different preprocessing methods and the output format of the resulting images. The resulting images serve as input to module 2 of the software.

Following the preprocessing, the detection for the photogrammetric markers is conducted. A single Machine-Learning Model accepts the preprocessed images of first software module as sole input. Performing object detection, the second module returns the coordinates of rectangular area containing the markers. Further the results comprise the predicted class of the marker (also "class-id"), as well as a probability score for a correct prediction. Fig. 3 superimposes the results of the second software module onto the original image. It is then passed on to the third software module.

In the third stage of our methodology, we determine the center of the markers and decode the point-numbers - if applicable. As mentioned before, at least six coded targets need to be detected and identified correctly. For uncoded markers, and instances where decoding of the point number is unsuccessful, continuous point numbers are assigned. As a final step in the process, the acquired information is written to the coordinate files.

1. The name of the image.
2. The point-numbers of the markers, alternatively a consecutive number.
3. The image-coordinates of the markers.
4. Additional information which comprises the probability of the ML-score, estimated accuracy and further geometric or statistic information.

4. Software design

The following subsections describes the applied methods in the three modules in detail.

[1] The high concentration of salt in ocean water lowers its freezing point from 0 °C to −2 °C. In addition, the high volume as well as tide and currents prevent the water molecules from freezing into the stationary state of ice crystals.

[2] The conversion from raw image to JPG does not significantly change the accuracy of the marker center detection. But it might lead to an increased detection rate, depending on the detector settings. See (Reznicek et al.).

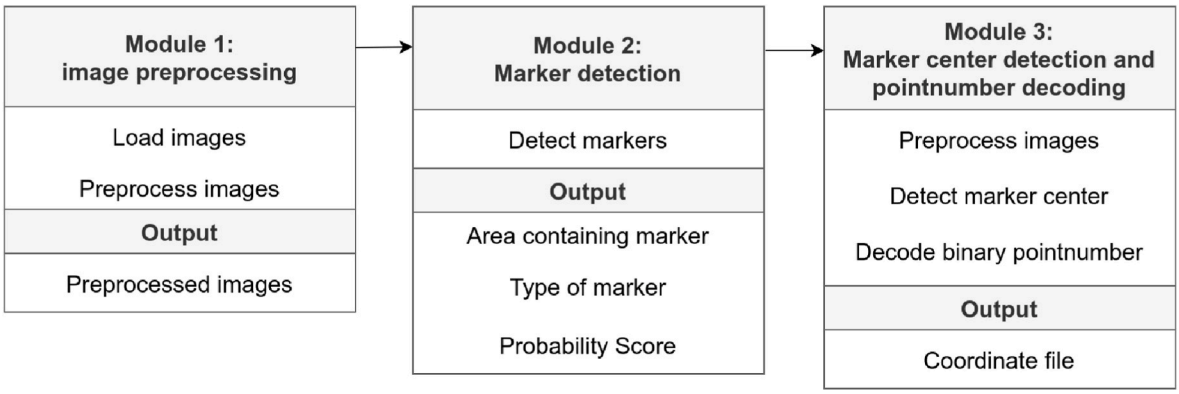


Fig. 2. The software design. The division into three modules serves an improved overview and enables a higher degree of flexibility.

Table 1
Output-format of the 4 preprocessing methods.

Model - No.	preprocessing method	Output format
1	None	full depth color image
2	CLAHE	grayscale image
3	Kirsch-Operator	grayscale image
4	3-Steps-Method	binary image

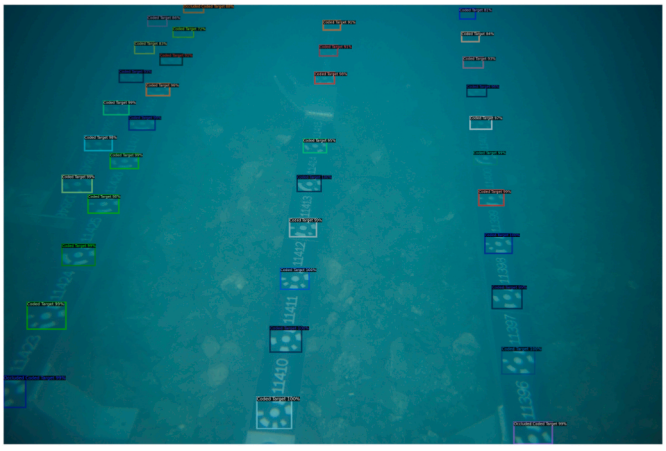


Fig. 3. Bounding-boxes, class and probability-score superposed on an original image. Haze appears frequently in underwater images.

4.1. Image preprocessing

In order to benchmark the performance of the various preprocessing methods, the training of the ML-model for marker detection is conducted four times. Every training session is fed the same image data set, but each time the images are preprocessed differently: 1. the original images (no preprocessing), 2. the contrast limited adaptive histogram equalization, 3. the Kirsch-filter and lastly 4. the 3-step processing chain with auto-parametrization. The following section presents the image processing methods.

The original images "as is", without any preprocessing method applied. Fig. 4 shows an example. **CLAHE** is an image processing technique developed to achieve a high contrast. The image is divided into small tiles, in which frequent intensity values are redistributed. This leads to an increased contrast within each tile. During re-assembly, the bilinear interpolation remove artificial artifacts at the boundaries of neighboring tiles, see Fig. 5.

The Kirsch-filter has proven to deliver good results in underwater images. It is well suited for the enforcement of radial edges. The kernel

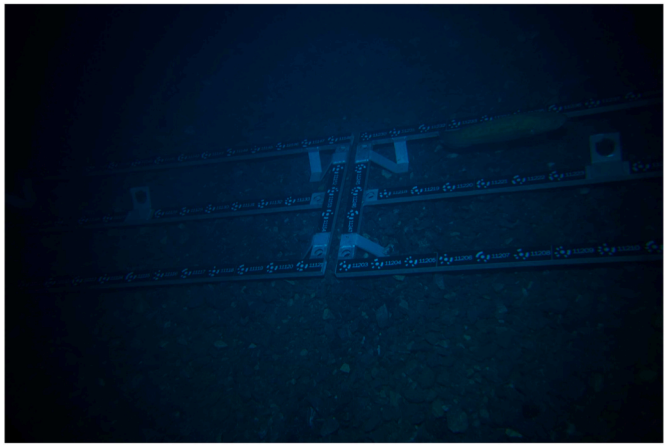


Fig. 4. Unprocessed, original image. The lack of light at the image borders is a typical feature, which appears like a strong vignette.

Fig. 5. CLAHE-processed gray-scale image. The enhanced contrast reveals additional markers close to the image border.

Table 2
Kernel mask of the Kirsch operator, western orientation.

5	5	5
−3	0	−3
−3	−3	−3

mask (see Table 2) is convoluted with the original image, and gets rotated according to the eight compass directions. The resulting edge direction is defined by the mask that produces the maximum edge magnitude. The mask is shifted over the entire image, repeating this process. Fig. 6 shows the image after the Kirsch filter is applied.

The 3-steps method is a combination of three processing methods. It has been developed, based on experience with image processing combined with empirical testing. Its primary goal is to enhance strong edges of the photogrammetric markers in order to maximize the marker detection rate. Secondary goal is to blur the background in order to reduce the amount of false positive detections. In a first step, we increase the contrast of the images using CLAHE. In a second step, we apply the bilateral filter, which is non-linear, edge-preserving, and noise-reducing. It smooths the image by replacing each pixel with a weighted average of its neighbors. The weight consists of a spatial component that penalizes distant pixels, as well as a range component that penalizes pixels with a different intensity. This combination ensures that nearby and similar intensities heavily contribute to the new pixels' value, while distant and different intensities do not contribute to the new pixels value. With this concept, the filter blurs areas with low contrast, such as background noise, while maintaining areas with high contrast, such as black-and-white markers. For reference, the bilateral filter is well explained in (Paris et al., 2008). As a third step, we apply the Kirsch filter. It enhances the edges that have not been blurred by the bilateral filter. The implementation within the 3-step method goes beyond the conventional Kirsch filter: an edge map along with their respective directions is generated using the Kirsch filter. Derivatives for each pixel are computed using predefined convolution masks to visually represent the edge directions of the detected edges. Fig. 8d depicts the image post-application of this implementation. Fig. 8d illustrates the effects of the procedure without the utilization of the bilateral filter. Another image processed with the 3-steps method is presented in Fig. 7.

Finally, we binarize the image based on a gray-value by manually setting threshold. This step further suppresses background information. Fig. 8 visualizes the effects of each of the processing steps. Exemplary results of the 3-steps method are shown in Figs. 8c and Fig. 7.

The 3-steps processing chain initially requires 7 parameters: The first three parameters are required by CLAHE: one for contrast limiting and two values for dividing the image into tiles. Further, three parameters are required for the bilateral filter: the diameter of the pixel neighborhood, and two parameters for the impact of the color space. The Kirsch filter, in this configuration, requires one parameter: the threshold for the maximum edge direction derivative. The parameters count may be reduced from seven to two: the three parameters for the CLAHE are chosen to be static, as empirical testing shows that this delivers satisfactory results for all images. Similarly, the bilateral filter receives static, yet low values for the euclidean distance and the color space. Small values ensure that high contrast areas are not blurred. In order to blur weak contrast areas nevertheless, the filter is applied consequentially several times. A single integer sets the number of iterations and is the first parameter that has to be defined. The approach results in a repetitive smoothing of low-contrast areas, while high-contrast-edges remain preserved throughout all iterations. Finally, we introduce a second parameter as threshold for the binarization of the image.

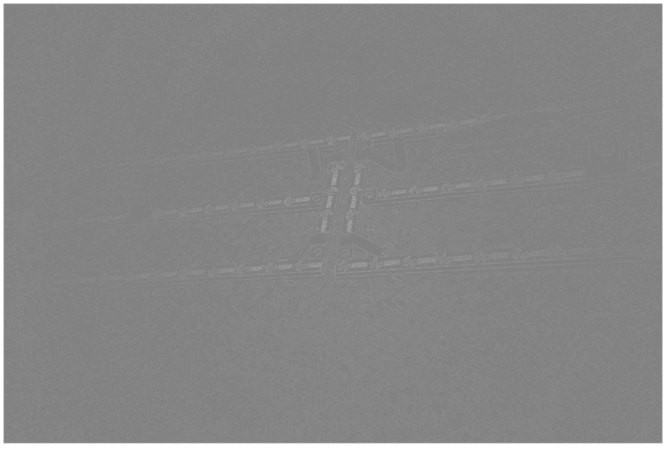


Fig. 6. Applying the Kirsch-filter, the resulting gray-scale image has a grayish haze, and strong edges are enforced. Depending on the print, the signal might be difficult to see.

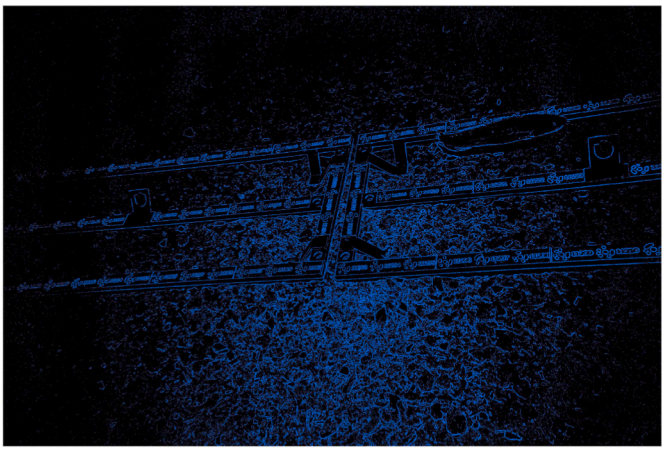


Fig. 7. Final results of the 3-step-processing method. Contrast-weak areas are suppressed as long as the contrast strong areas remain unaltered. Please observe the markers in close to the image borders.

These two parameters are to be set by a trained ML-model. We use 297 images as ground truth for supervised training. The ground truth is generated by processing the image data-set with the 3-steps method. We manually specify the two parameters by obeying the following rules: firstly, we choose the parameter for the binarization in order to maximize the amount of markers that get reinforced by the Kirsch operator. Secondly, we maximize the number of iterations of the bilateral filter, and maximize smoothing of the background - as long as the first rule is not violated.

The model is based on the "very deep convolutional network" from the "Visual Geometry Group"[3] (short: "VGG"). The VGG network is an optimized model for object recognition, supporting up to 19 layers - see (Simonyan and Zisserman, 2014). For the described task, 11 layers have proven to be efficient. The model takes the image at the input layer. The last layer is adjusted to output two values, resulting in the two parameters for the 3-steps method.

Supervised training is employed, utilizing a dataset of 297 images along with the ground truth in a CSV file, which provides both parameters in normalized form for each image. The training dataset comprises 85 %, the verification data set 15 %. These values are manually determined by initially estimating the parameters and then optimizing them through multiple iterations. Two rules are followed during this process: firstly, the threshold for binarization is chosen to ensure no signal from the markers is suppressed. Secondly, the number of iterations of the bilateral filter (and thus, blurring of the background) is maximized, as long as the first rule is not violated. The resolution of the images is reduced to a maximum of 512 pixels on each side to meet the VGG network requirements. Further, the imagery is artificially increased by applying random crop. An example of an image the model produced as result is shown in Fig. 7.

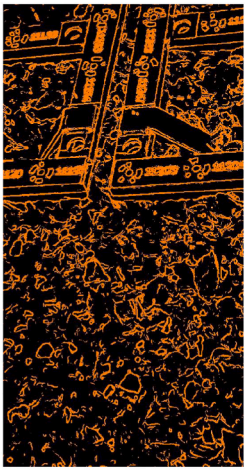
As the optimization function for weight adjustments, the widely used ADAM optimizer (Kingma and Ba, 2014) is employed. Adam is a stochastic gradient descent method based on the adaptive estimation of

[3] Faculty of Engineering, University of Oxford.

(a) CLAHE results in a high-contrast gray-scale image.

(b) The bilateral filter blurs areas with low contrast, while preserving areas with high contrast.

(c) The Kirsch and the thresholding result in a binary edge image.



(d) For comparison: this image skips the bilateral filter, resulting in a binary edge image with significantly increased background information.

Fig. 8. These example images visualize the impact of the methods of the 3-steps-method.

first and second-order moments, calculating a separate learning rate for each parameter. The loss function used is MSELoss, and the learning rate scheduler used is "ReduceLROnPlateau",[4] which reduces the learning rate based on the validation loss during the training process.

4.2. Marker detection

Taking advantage of the machine learning capabilities of object detection, we detect photogrammetric markers within the second module of our implementation. The model predicts the location of the marker by returning a bounding-box for each prediction. It also returns the type of marker as well as the probability-score. We apply Detectron2 (Wu et al., 2019), Facebook AI Research's library which provides state-of-the-art detection algorithms. It is written in Python and bases on PyTorch. We customize and re-train a pre-trained object detection model from the Detectron2 model zoo, instead of creating our own model from scratch. We choose the latest and best performing model, which is at the time of writing the "faster_rcnn_X_101_32 × 8d_FPN_3x". It is based on faster region proposal networks ("R-CNN") (Ren et al., 2015) with ResNet-101 as backbone. During the training process, each session produces a model, which is saved. We determine the best model by saving the current model as "best model", if its loss is less than the previous one. In the ground-truth data set used for supervised the training, we distinguish between 4 types of markers. The distinction allows to develop specific algorithms for each class.

1. Coded targets: The targets developed by Schneider (Schneider, 1991) are chosen as coded targets, as they come with a low degree of detail and perform well in terms of location and accuracy in poor visibility conditions (Drap et al., 2013). The markers are printed in two sizes, with white markers on black background. The 20 bits binary code chosen enables rotation invariance.
2. Uncoded targets: Simple circular targets. The markers are printed in two sizes, with white marker on black background as well as yellow marker on black background. In contrast to the coded targets, the circle representing the center is larger in the majority of cases, making the makers easier to detect and resulting in more accurate center-coordinates. Uncoded targets do not come with a unique point-number.
3. Occluded coded targets. These are coded targets that are only partially visible. Common reasons for occlusion are: the marker is partially outside the image border, it is partially hidden behind a fish or partially covered by sand. We define that poor visibility does not lead to the classification "occluded".
4. Occluded uncoded target: Identical to occluded coded targets, except that the markers do not have binary codes.

Fig. 9 presents an example of each marker type.

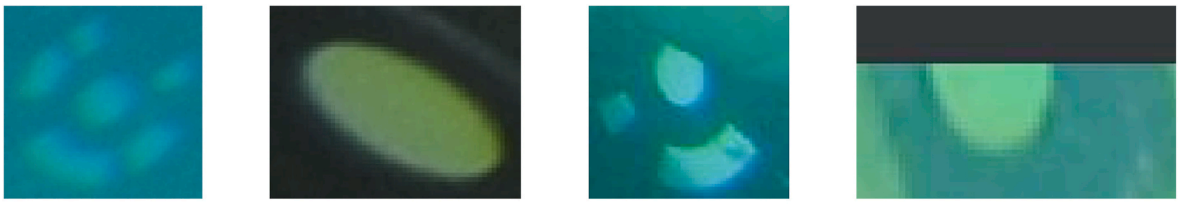
The ground truth imagery for the supervised training comprises 7726 annotated images. These images contain 336,202 bounding boxes along with the corresponding marker-type. During the annotation, a high standard of quality is ensured by involving two independent labeling teams, both of them having internal quality control measures in place. The images are not preprocessed for the purpose of labeling. As labeling software, "LabelImg"[5] and "COCO-Annotator"[6] are applied. Table 3 shows how the data is split between the three data sets for training, testing and validation. About 80 % of the images are used for training, about 20 % were used for testing and validation.

Fig. 10 visualizes the distribution of the classes within each data set. There is a significant bias towards coded targets in all data sets. In contrast, occluded uncoded targets are strongly underrepresented. This is to be expected as the data set consists of real-world data where 1) coded targets are deployed whenever possible. 2) Uncoded targets are deployed when the scene lacks space for coded targets. 3) Both types of occluded targets are avoided whenever possible. The member-distribution in the train-data set differs significantly from the test- and validation data sets, which underlines their independence during testing

[4] https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.ReduceLROnPlateau.html.

[5] lableImg as become a part of the Label Studio community: https://github.com/HumanSignal/labelImg.

[6] https://github.com/jsbroks/coco-annotator.



(a) Coded target.

(b) Uncoded target. The image is acquired with a strong angle, and medium contrast.

(c) Coded target, partially occluded by a fish.

(d) Uncoded target, only partially recorded due to the image border.

Fig. 9. The object detection model is to distinguish between 4 different type of markers.

Table 3
The created ground truth data and classification.

Data set	Amount of images	Total amount of bounding boxes
Train	6181	223660
Test	775	43116
Validate	770	69426
Total	7726	33602

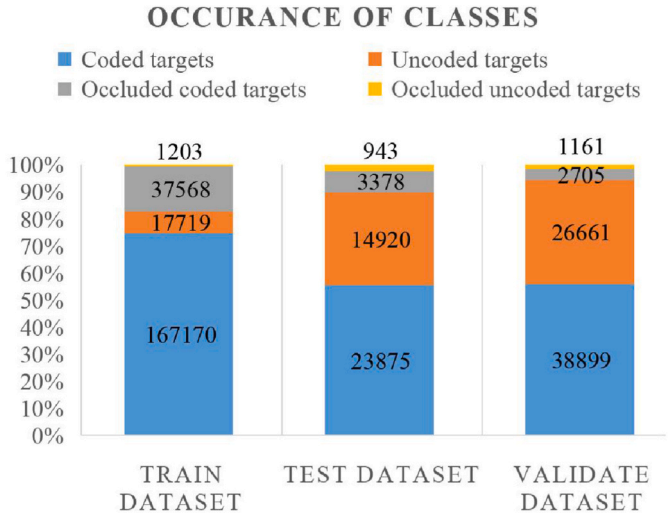


Fig. 10. Occurrence of classes within the three data sets. A significant class imbalance is present.

and validation.

4.3. Center detection and binary code decoding

The third and last step of our approach is implemented in the third module: its task is to detect the center of the marker and - if applicable - to decode the binary code. As input, the module receives the bounding boxes as well as the maker class from the previous module. Both, the center coordinates and the binary code are to be stored in text files. Each of the steps are described in the following paragraphs.

The center of the circular markers is represented by the center of a fitted ellipse. The developed process is alike for coded- and uncoded targets. In case of coded targets, we firstly distinguish between "center feature" and "binary code features" within the bounding boxes. This differentiation is conducted by a neural network, as conventional methods such as blob-detectors often falsely classify small 20-bits segments as center feature. As presented in Fig. 11, the resulting ML-model creates a single bounding box, surrounding the center feature. Any pixel outside the center feature is part of the binary code feature. A ground truth of about 1600 images was used for training in order to achieve a high success rate. Once again, we apply Detectron2. The configuration is identical to the one for the detection of targets.

Once the inner circle of the targets is isolated as shown in Fig. 11b, we apply single-level thresholding. This is an effective method when treating a small region of an image, in which constant contrast can be assumed. Empirical testing has revealed, that none of the common thresholding methods leads to satisfactory results for all visibility conditions. However, amalgamating the methods of Yen, Otsu, Max. Entropy and Huang (Yen et al., 1995; Otsu, 1979; Sahoo et al., 1988; Huang and Wang, 1995) has proven to work well. We combine these four thresholding methods pixel-by-pixel: the resulting pixel will only be white if all four binary images show the pixel as white; Fig. 12 shows an example.

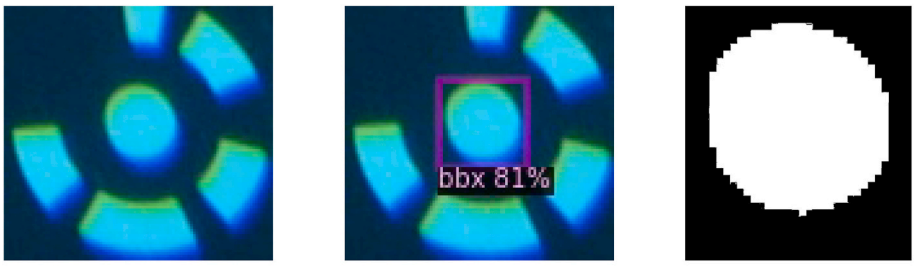
We create a mask by fitting an ellipse to the thresholded gray-scale image. This ensures that only relevant pixels contribute to the following computation of the marker center. The center is represented by the image moment, the weighted average of the pixels intensities (see formula 1).

$$MomentsM_{ij} = \sum_x \sum_y (x - \bar{x})^{i} * (y - \bar{y})^{j} * I(x,y), \quad (1)$$

where x and y define the row and column, I(x,y) refers to the intensity of the pixel, and i, j refer to the order. The moment of the center feature is saved with sub-pixel accuracy as center of the marker.

In the next step, the decoding of coded targets is performed. Subsequent, the binary code is to be converted into the corresponding decimal number, which represents the point-number of the marker. Finally, the center-coordinates and the point-numbers are exported into coordinate-files, ready for bundle adjustment.

Twenty-bit codes are challenging to decode, especially when the markers are close to the image border, where distortion and poor light conditions are present. As a first step, we improve the geometric center of the bounding box provided by the ML-model in module 2, replacing it with the center-coordinates of the marker. This compensates for potentially low geometric accuracy of the ML-model. Next, we prepare the images for the subsequent geometric analysis: we crop from the original image and apply CLAHE. We binarize as described in the previous section and increase the resolution of the image segment by factor 10. We search for closed contours and treat them as potential 20 bits segments. Two measures are applied to reject false positives: firstly, we filter the results by size. Contours that are too small for twenty-bit segments are rejected. Secondly, we reject contours with a distance to the center-point that exceeds half of the image segment size. Both measures help to remove contours that have been falsely categorized as part of the marker during the binarization process. Finally, we compute an ellipse around the center point. A 4-point-transformation rectifies the ellipse, resulting in a more circular code. We draw an increased ellipse through the center of the binary code. The ellipse is partitioned into 20 segments with equal angles. Finally, the extracted binary code is



(a) The detected marker.

(b) The detected center feature of the marker.

(c) The binarized center feature serves as a mask. Applied to the grayscale image, it excludes unwanted pixels from the computation of the marker center.

Fig. 11. The process of the marker center detection.

(a) Marker in original image. The human eye can hardly detect any signal

(b) Marker after binarization, combining 4 popular thresholding methods. The geometry of the marker is restored.

Fig. 12. Binarization of markers combining 4 different thresholding methods.

converted into a decimal point-numbers. Since the binary codes lack start- and stop-bits, this step requires a convention. We offer the user to choose between two conventions that are implemented in form of lookup-tables: the Agisoft Metashape convention and the Hexagon Aicon convention. Any other convention may easily be integrated.

5. Performance assessment

The performance of the present approach is measured in two ways: we will conduct a statistical comparison of the fourfold trained neural network applying an object detection metrics. Second, we set the commercial software "Metashape professional" from Agisoft and our prototype "remote Photogrammetry", aka "rePho" to detect and identify the markers in a range of images, comparing their success rates.

5.1. Setup of the statistical evaluation of image preprocessing

This section presents the performance measurement of four object detection models derived from four different image preprocessing methods. The detections are evaluated following the methodology of the PASCAL Challenge (Everingham et al., 2010). We employ the open-source tool developed by Padilla et al. (2021).

The output of an object detector is characterized by a bounding box, the class, and a confidence score. All three measures are evaluated by the AP, which compares the predictions of the model with the ground truth.

1. Classes: we distinguish between the four classes "coded target", "uncoded target", "occluded coded target", and "occluded uncoded

target". The prediction may be considered as True Positive ("TP") only if the predicted class matches the ground-truth class,

2. Bounding box: we compare the area of a predicted bounding box (in image pixels) with the ground truth annotation by defining the intersection of union ("IOU"). The IOU is defined as the overlap of the predicted bounding box and the ground truth, divided through the area of union:

$$Intersection\,of\,Union = \frac{Area\,of\,Overlap}{Area\,of\,Union} = \quad (2)$$

If the confidence $>$ 50 %, the detection is considered as TP. As a precondition, the predicted class has to match the class of the ground truth - otherwise, it is characterized as False Positive (FP).

3. Confidence score: the confidence score is the probability threshold of the marker being detected correctly. This value is given in percentage.

The confidence score is taken into account by accepting predictions as TP only if the confidence is larger than the confidence threshold. This threshold is introduced during the computation of the precision and the recall. The precision is the ability of a model to identify only relevant objects - in other words: precision is the percentage of correct positive predictions among all detections:

$$Precision = \frac{\sum True\ Positives}{All\ Detections} \quad (3)$$

The recall is the ability of a model to find all relevant cases - in other words: the percentage of correct positive predictions among all given ground truths.

$$Recall = \frac{\sum True\ Positives}{All\ Ground\ Truths} \quad (4)$$

The AP is defined as the area under the precision-recall curve. It summarizes this precision-recall trade-off dictated by confidence levels of the predicted bounding boxes. The mAP represents the exactness of the detections among all classes. It is the simple average of all classes:

$$mAP = \frac{1}{C}\sum_{i=1}^{C} AP_i \quad (5)$$

where AP_i is the AP value for the i-th class and C is the total number of classes (here: four) being evaluated.

5.2. Results of the statistical evaluation of image preprocessing

All four trained models are set to process the 770 images of the validation data set. A total of 69,426 bounding boxes build the ground truth. The subdivision is shown in Table 4.

The IOU threshold is set to t = 0.5 in order to classify detections as TP or FP, We deliberately choose a rather low threshold since a moderate geometric detection accuracy may still lead to a precise location of the marker as well as the correctly decoded point-number, see 'centering of the bounding box' at the end of chapter 4.3. The minimum confidence score is set to 0.85. Fig. 13 shows the precision-recall-graphs for all four trained detection models, comprising all four classes.

Fig. 14 depicts the Average Precision (AP) using the PASCAL VOC metric as a block diagram. The Mean Average Precision (mAP) for the four ML models is presented in Table 5.

Table 4
The distribution of bounding boxes per class within the ground truth. Note the significant class imbalance.

Coded Targets	Uncoded Targets	Occluded Coded Targets	Occluded Uncoded Targets
38,899	26,661	2705	1161

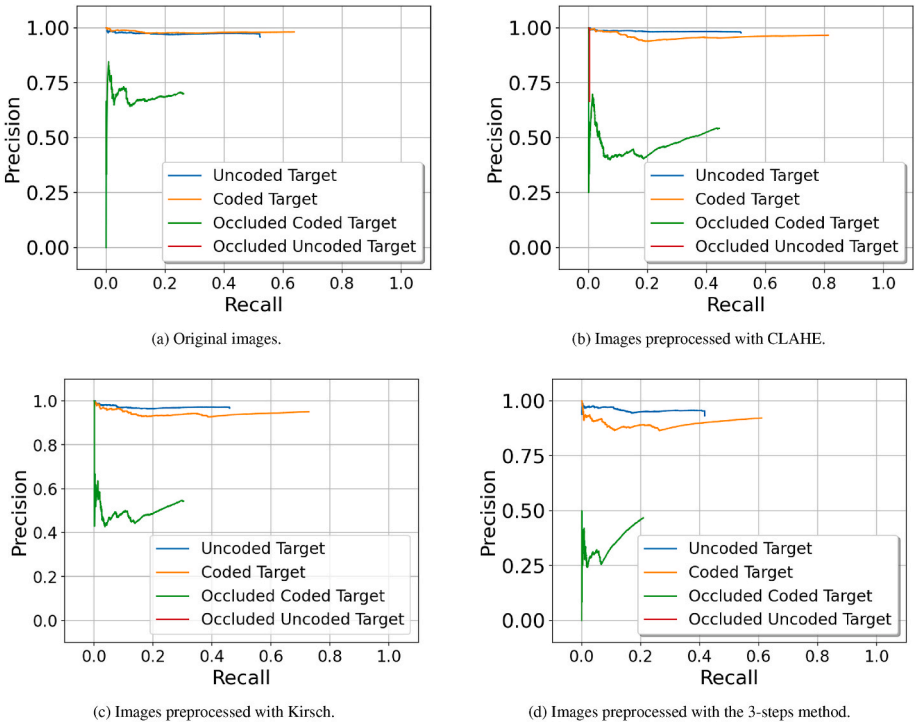
Evaluating the previous figures, we can conclude that the CLAHE-treated images result in a superior performance of the trained ML-models. In total, CLAHE detects slightly over 70 % of all markers in the ground truth within the validation data set. It is the only model that manages to detect some of the rarely-appearing occluded uncoded targets, which account for only 5 ‰ of all markers in the training data set. While the model trained with the original images and the model trained with edge-images perform similarly with a mAP of 0.33, the 3-steps method clearly falls behind and achieves only a mAP of 0.27. Fig. 15 shows the number of TP's within each of the four classes for the four models, as well as the ground truth.

5.2.1. Comparison with commercial software

We process 26 selected images with our prototype rePho and the commercial software Metashape professional from the company Agisoft. Metashape is chosen as it is considered to be the most performant software for the detection of coded targets. Our goal is to determine the softwares capabilities of detecting and identifying photogrammetric targets in underwater images. The images reflect a broad range of typical challenges - ranging from good visibility to extreme darkness, strong backscatter, haze, dirt, shadows, and strong chromatic aberration. At least two neighboring images were chosen since Metashape's workflow requests at least two images to be aligned, before the feature for detection of uncoded targets is unlocked. If the alignment fails, no uncoded targets may be detected. The following list summarizes the image properties.

- Images 1–4 are taken in good visibility, and, therefore, at a great distance, which results in small markers. Object- and image plane are under a small angle. The water causes a blueish color and a strong chromatic aberration at the marker-contours is observed.
- Images 5 and 6 are similar to the previous ones, but are taken with about 45° angle between object- and image plane. Consequently, the markers gradually darken towards the further image border due to the increasing distance. The effect is moderate. These images as well appear blueish.
- Images 7 and 8 are greenish, with some haze and a thin layer of dirt covering some of the markers. The distance and angle between camera and markers is about 15°. See Fig. 16.
- Images 9 and 10 suffer from strong haze, and have therefore been taken at very close range. Consequently, the markers appear large. The camera aims perpendicular to the object plane, and the resulting image is quite green. See Fig. 17.
- Images 11 to 18 suffer from severe darkness, which causes strong noise. The camera is quite close to the markers, with moderate angles - about 15–30°. In some of the images, a little backscatter can be identified, as well as moderate chromatic aberration. See Fig. 18.
- Images 19 and 20 are similar to the first images, but have been taken with an angle about 30 °between camera and object plane. Due to the increased distance, some of the markers are dark.
- Images 21–24 suffer from severe backscatter, due to the camera was again positioned quite close to the markers. Nevertheless, the images are very blurry and noisy. The camera points perpendicular to the object plane.
- The last two images (25 and 26) are taken once more at close distance, with about a 15 °angle between camera and object plane. The images are greenish, and the markers are covered by a significant amount of dirt. See Fig. 19.

rePho receives the original (unprocessed) images as input, and applies CLAHE as part of its internal workflow. In contrast, Metashape receives two sets of images: the originals, as well as the images with CLAHE applied. RePho processes all images without any setup or interaction on behalf of the user. In contrast, the workflow of Metashape



(a) Original images.

(b) Images preprocessed with CLAHE.

(c) Images preprocessed with Kirsch.

(d) Images preprocessed with the 3-steps method.

Fig. 13. The precision-recall-curves for all four marker classes. All values in percentage (1 corresponds to 100 %).

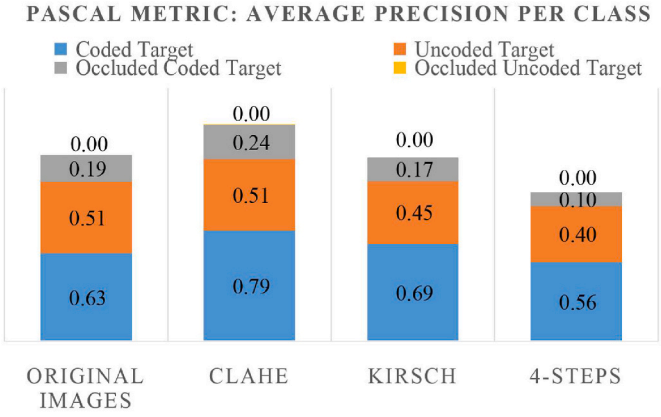


Fig. 14. The AP per class for the four trained models, applying the Pascal Metric.

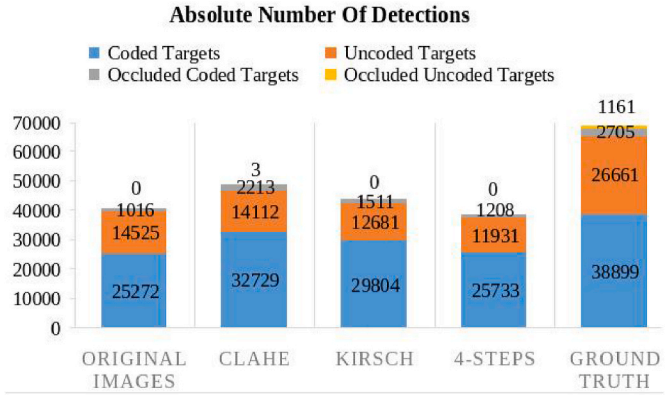


Fig. 15. Comparison of absolute number of TP across all four trained models, as well as the ground truth. Note that only the model trained with CLAHE preprocessed images was capable to detect occluded uncoded targets.

Table 5
The mAP is defined as the arithmetic mean of the AP calculated across the four classes. Each of the four classes is weighted equal, the class imbalance is ignored.

Original Images	CLAHE	Kirsch	3-Steps
0.33	0.39	0.33	0.27

requires both: the tolerance threshold for the detection of coded target is set to 100 %. If reduced, less (or no) targets are detected. The required alignment between images fails in cases where too few coded targets are detected or too many errors during the decoding of the point number occur. This could be compensated by manual target-picking, which has not been performed. Consequently, in theses cases, Metashape does not detect any uncoded targets. The tolerance for the uncoded targets is adjusted to 50–70 % in order to balance between too many false-positives and too few detection's. Fig. 20 compares the number of correctly detected and identified coded targets within each image. Fig. 21 compares the number of correctly detected uncoded targets. Fig. 22 compares the number of uncoded targets, with correct center location and incorrect decoded point number, while Fig. 23 shows the amount of coded targets with correct center location, where no point number could be retrieved. For these cases, rePho provides a random point number in order to retain the valuable information of the center coordinates. These points are then stored as uncoded targets, which allows us to use them at later stage during the bundle adjustment. Lastly, Fig. 24 provides an overview of the comparison.

5.3. Summary and discussion

Referring to Fig. 15, we can conclude, that the model trained with CLAHE-preprocessed images has the best overall performance in the validation data set. It achieves a mAP of 0.39. The model succeeded in detecting 84.1 % of the coded targets, 52.9 % of the uncoded targets, 81.8 % of the occluded coded targets and 0.3 % of the occluded uncoded

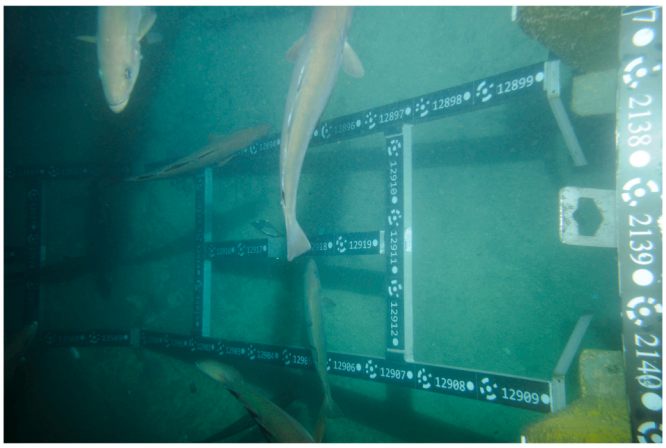


Fig. 16. Greenish example image with strong angle between the image- and the marker-plane. Some haze and some shadows are present.

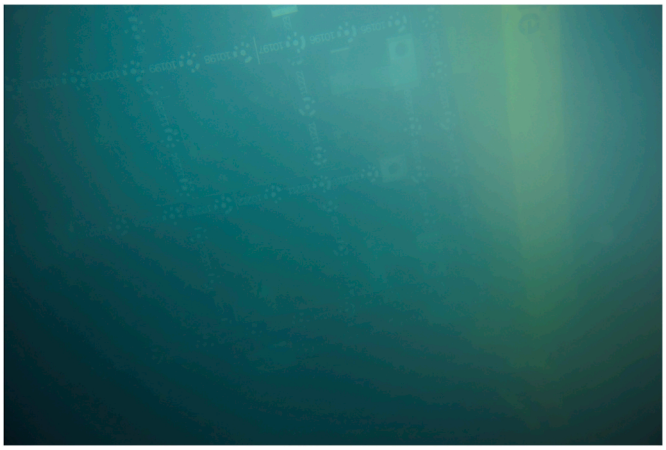


Fig. 17. Example image with strong haze.

Fig. 18. Very dark example image, taken at close distance.

targets. The ranking corresponds with the ranking of the number of members of each class in the training data set.

The relatively low success rate of uncoded targets detections may be improved by extending the training data set with a high number of class members. While this is likely to succeed, increasing the amount of occluded coded targets is more challenging: artificially cropping images in order create more occluded targets will in fact increase the absolute

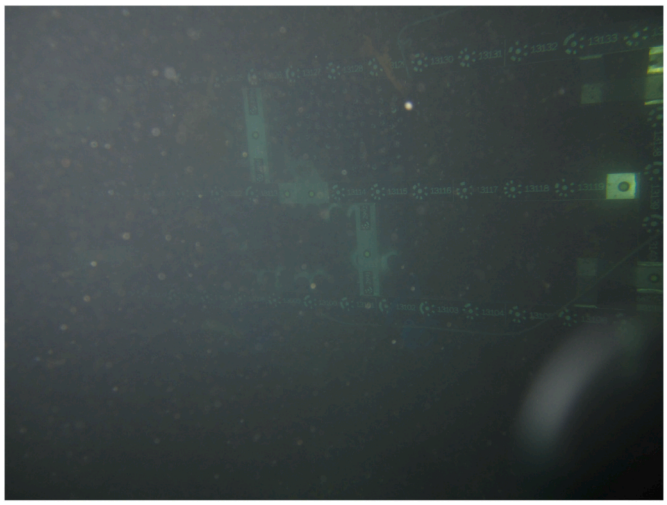


Fig. 19. This example images shows strong backscatter effects.

amount of occluded targets. But it will not decrease the class imbalance since the amount of non occluded targets will in almost all cases be higher. As an alternative, artificial data could be created using image augmentation. However, the detection of occluded targets has a low priority, since there is little to gain, compared to an performance improvement on the non occluded targets.

We also conclude, that the overall performance of the 3-steps method does not lead to the desired outcome. The mAP ranks lowest in the comparison at 0.27. The processing chain detects 66.2 % of the coded targets, 44.8 % of the uncoded targets, 44.7 % of the occluded coded targets and 0 % of the occluded uncoded targets. One reason may be that the bilateral filter and the binarization inevitably remove some of the low-contrast markers in the output image. In these cases, the annotation data indicates the presence of a marker, while there is actually no signal within the given bounding box. This will lead to contra-productive training. A solution to this would be to verify and edit the annotations of the ground truth.

Further, the results confirm that the number of members within each of the 4 classes has a significant impact on the performance of the detection rate of the ML-models. The measures that were taken to fight class imbalance during training, succeeded to a certain degree when it comes to the coded targets. Occluded targets however, remain at a low AP (or even zero).

The comparison between rePho and the software Metashape reveals, that rePho detects and identifies more coded targets correctly than the two competing setups. Metashape succeeds in 41 %, “Metashape CLAHE” in 53 %, and rePho in 58 % of the cases. See the first chart in Fig. 24. Decoding 20 bit codes in underwater images is very challenging. Yet, this task is a crucial benchmark: image orientation and, consequently, the renaming of continuous point numbers of the uncoded targets depend on it. For a first image orientation, a minimum of 3 correctly decoded coded targets are required ((Luhmann et al., 2006), chapter 4.2.3.2). Considering requirements, e.g., regarding the geometric distribution of the markers within the images, as well as the goal to automate the entire process, at least 6 coded targets should be detected and identified correctly. Looking closer at Fig. 20, Metashape CLAHE achieves superior results compared to rePho in the images 11–18, which suffer from severe darkness and noise. Future training data for rePho should include comparable images in order to improve the performance. In contrast, Metashape using the original images fails to pass the defined threshold in 8 of all 36 images, which makes an automated process unlikely to succeed.

The second chart in Fig. 24 shows the number of incorrectly decoded point numbers - a crucial benchmark, since a high percentage will result in the disabling of automation for the bundle adjustment. Metashape

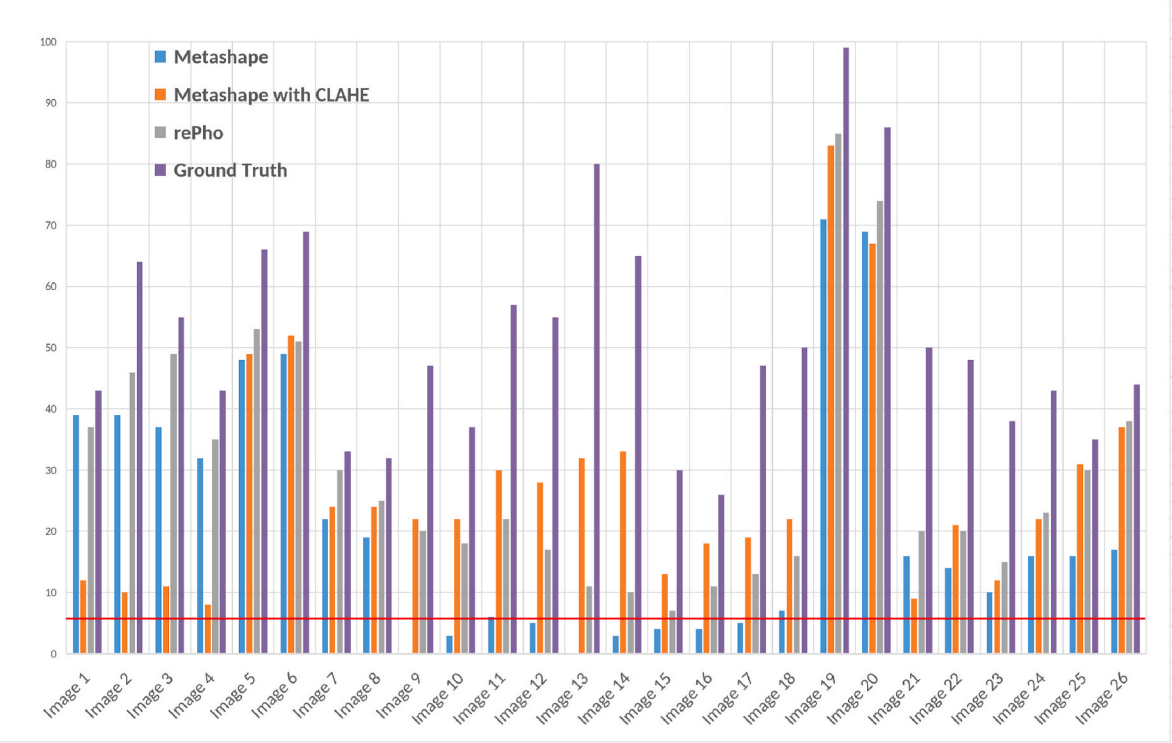


Fig. 20. The number of detected coded targets with correct center location and correctly decoded point number. The target threshold of 6 detections is drawn in red. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

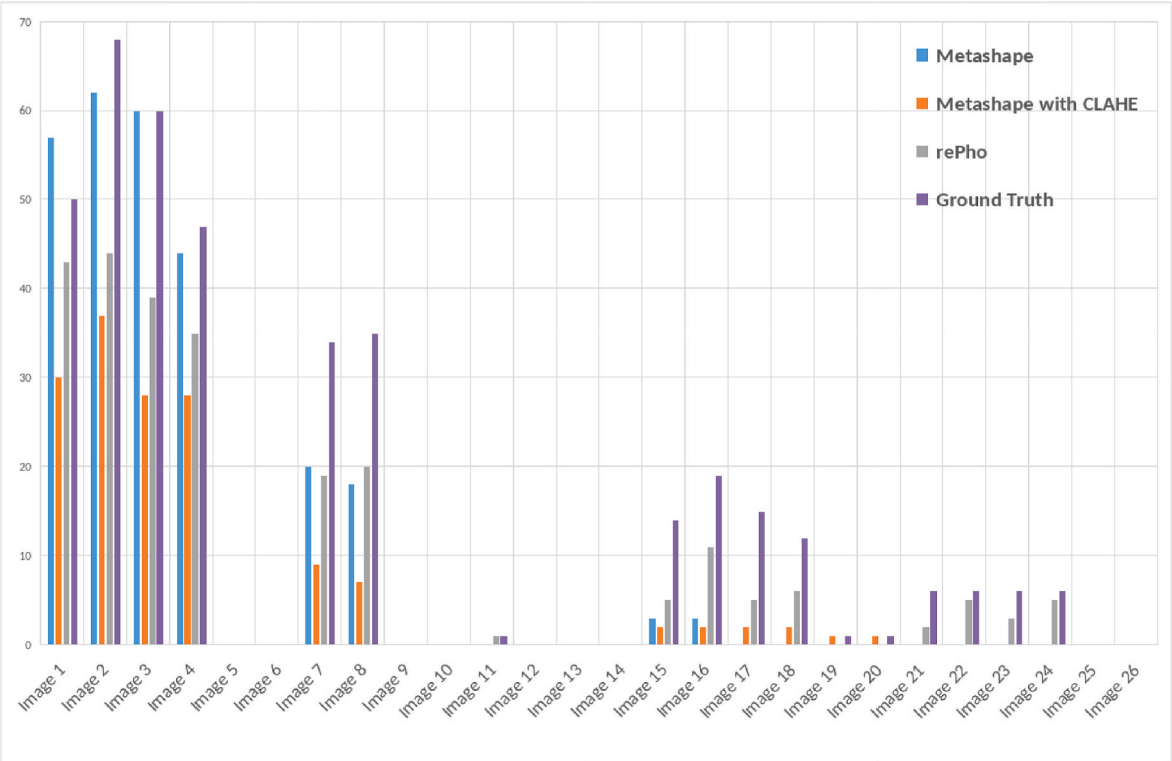


Fig. 21. The number of detected uncoded targets with correct center location.

fails in 35 cases (6 %), Metashape CLAHE fails in 152 (21 %), and rePho fails in 53 (7 %) cases. Fig. 24 shows rePho's capability to convert coded targets into uncoded targets: if decoding of a point number fails, and instead a unique point number is provided, a marker may still contribute to the bundle adjustment. In contrast, Metashape classifies coded targets as uncoded targets by mistake: it simply fails to recognize the binary code, and classifies the inner circle as uncoded target. Metashape conducts this in 41 cases, Metashape CLAHE in 19 cases, and rePho in 356

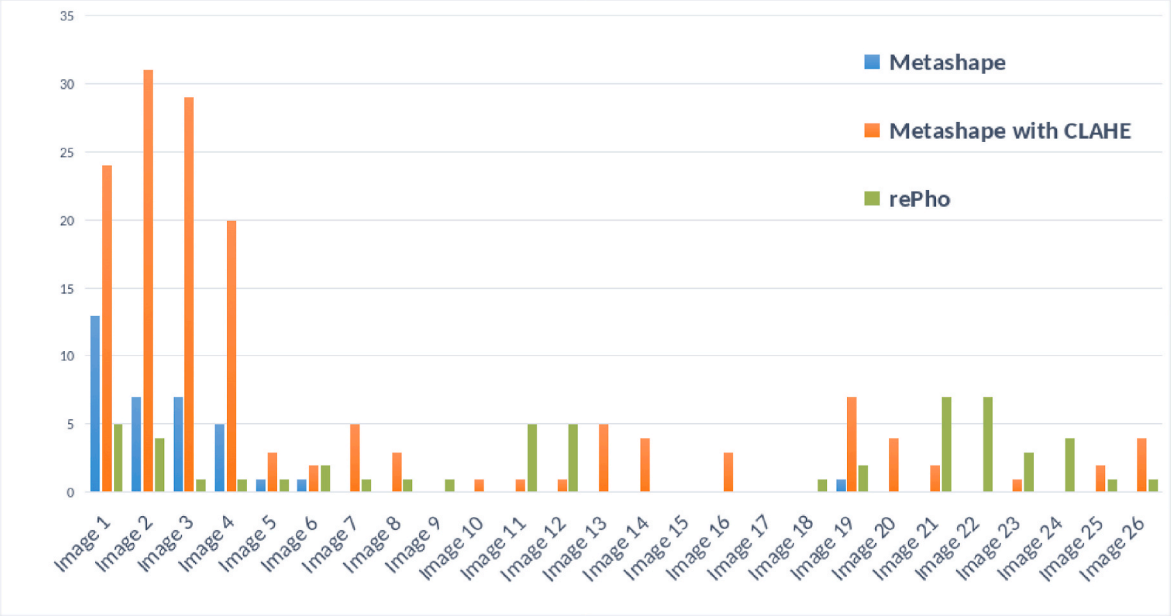


Fig. 22. The number of correctly detected coded targets with incorrect point number due to wrong decoding of the binary code.

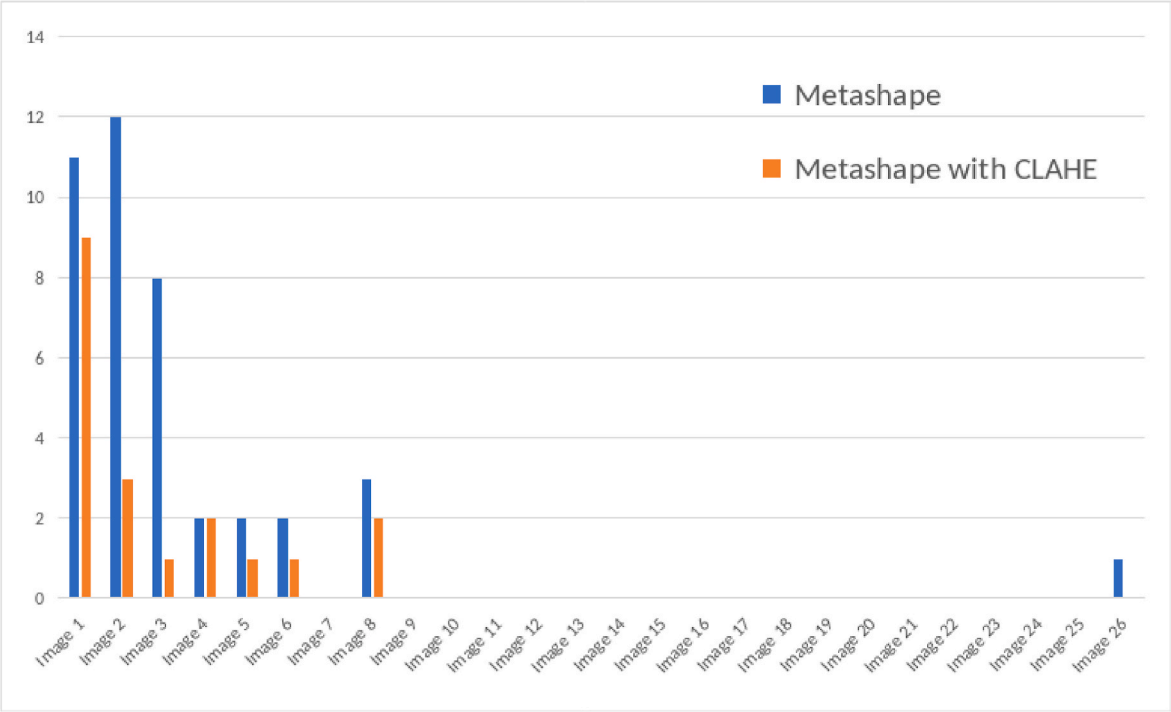


Fig. 23. The number of correctly detected coded targets that got successfully converted into uncoded targets because the decoding of the point number failed.

cases. Just as important is the correct detection of uncoded targets: Metashape succeeds in 267 cases (70 %) Metashape CLAHE succeeds in 149 (39 %), and rePho in 243 (64 %) of the cases. Please see the forth chart in Fig. 24.

A direct comparison between Metashape receiving original images and Metashape receiving images preprocessed with CLAHE reveals that the preprocessing has a strong negative effect on the detection of coded targets in the first four images. In contrast, the preprocessing significantly increases Metashape performance in almost all other cases (specifically in images 9–18 and 25–26). It appears that CLAHE has a negative impact when the measurement distance is large (images 1–4), where the overall visibility is good. In contrast, CLAHE seems to improves the performance at close measurement range, which is the case in poor visibility conditions - due to darkness, haze, etc.). The detection rate of uncoded targets decreases in most cases when image preprocessing is applied. Once again, the first four images, as well as images 7 and 8, show the strongest difference. CLAHE leads to fewer coded targets being classified as uncoded targets.

Image 15 represents rePho's worst case, in which only 7 markers are detected and decoded correctly. Nevertheless, this fulfills the requirement to detect and identify at least 6 markers per image. It is likely to lead to a correct renaming of 15 uncoded targets in the image (see

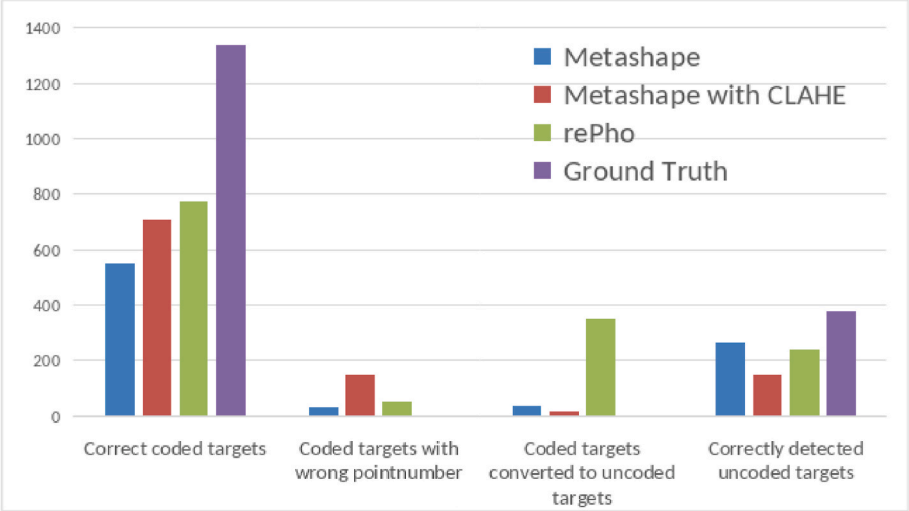


Fig. 24. Summary of comparison between 3 candidates and (if applicable) the ground truth.

Fig. 23), as no wrong point numbers enter the bundle adjustment (Fig. 22). These numbers are like to enable automation. On top, 5 uncoded targets will be renamed based on back projection and proximity (see Fig. 21). Metashape CLAHE succeeds as well to overcome the threshold. But in the first 4 images, only 8–12 targets receive a correct point number, while 20–24 targets receive a wrong point number. These cases will require user interaction during the bundle adjustment. RePho also detects 78 occluded targets, but wrongly classifies 26 targets as occluded. These markers are currently ignored when it comes to determining the exact center and decoding the point number.

6. Conclusion and future work

The proposed pipeline for the detection and identification of circular fiducial markers works reliably in the harsh underwater environments. This includes typical challenges like darkness, noise, dirt, haze and backscatter. We retrieve sufficient information from any of the tested images in order to conduct the following bundle-adjustment, without the need for the original images. The neural networks are trained with real-world image-data and have proven to work robust. The developed software processes large number of images reliably.

Our software "rePho" allows the user to choose between two different point-number conventions (Metashape- and Aicon-convention). RePho supports two types of export formats: image. dat and a basic text file for each image, with point-number, x-and y-coordinate. Additional information, such as indicators for reliability of the detected marker-class and for the accuracy of the ellipse center, is optionally included. In cases where the decoding of coded targets fails, rePho provides a continuous point number. This way, the bundle-adjustment may profit from the detected marker center and renumber the point numbers. The robustness of our models today delivers superior results compared to state-of-the-art commercial software. The software masters any of the tested visibility conditions.

The file size of a coordinate-file containing the information for all markers within a single image is about 1 kilobyte. A project with, e.g., 1500 images is thereby reduced from about 9 Gigabytes to about 1.5 Megabytes. This enables a new workflow, where photogrammetry-teams are split into field-workers and data-processing-specialists. The first part of the team focuses on the field work only. The latter may stay in the office and receives the coordinate files e.g. by satellite connection. This allows prompt data processing and timely delivery of the results, while simultaneously minimizing the amount of offshore workers required. Commercially available software like Metashape or Aicon (supplier: Hexagon) do not support this type of workflow, nor would the detection- and identification performance suffice.

Future work may include.

- Development of specific algorithms for the localization and identification of partially occluded markers.
- Improvement of the success-rate of the marker-detection model by increasing the versatility of the training data set. Special focus should be set on dark and noisy images, as well as uncoded targets. This may be achieved by annotating images where the models perform poorly, and using them as training data.
- Evaluation of the accuracy performance and correction of the eccentricity offset for the marker center. This is applicable for markers recorded with an angle between object-plane and image-plane of 45 deg. and above (Luhmann, 2014).
- In the field, images are recorded approximately every 5 s. The PC that was used during the development comprises an AMD Ryzen 9 3900X 2-threads 12-Core Processor and a NVIDIA GeForce RTX 3070 graphic card, which is utilized by the ML-model. Processing a single image requires about 40 s, if a high amount of targets are found. When processing multiple images at the same time, rePho takes advantage of multi-threading: 24 or less images are processed in about 50–60 s. This computation time should be reduced to less than 5 s for a single image, in order to make rePho realtime capable. There is a large potential in code optimization and updating the hardware.

CRediT authorship contribution statement

Jost Wittmann: Writing – review & editing, Writing – original draft, Validation, Software, Project administration, Methodology, Data curation, Conceptualization. **Sangam Chatterjee:** Supervision. **Thomas Sure:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work has been funded through the program "nærings-ph.d" by the Norwegian Research Counsel and the company "Elleve AS", Stavanger, Norway.

References

ARToolworks, Artoolkitx. URL http://www.artoolkitx.org.

Cejka, J., Bruno, F., Skarlatos, D., Liarokapis, F., 2019. Detecting square markers in underwater environments. Rem. Sens. 11 https://doi.org/10.3390/rs11040459.

Dosil, R., Pardo, X.M., Fdez-Vidal, X.R., García-Díaz, A., Leborán, V., 2013. A new radial symmetry measure applied to photogrammetry. Pattern Anal. Appl. 16, 637–646. https://doi.org/10.1007/s10044-012-0281-y.

Drap, P., Merad, D., Mahiddine, A., Seinturier, J., Gerenton, P., Peloso, D., Boï, P.-M., Bianchimani, O., Garrabou, J., 2013. Automating the measurement of red coral in situ using underwater photogrammetry and coded targets. ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XL-5/W2, 231–236. https://doi.org/10.5194/isprsarchives-xl-5-w2-231-2013. URL. https://github.com/facebookresearch/detectron2.

Everingham, M., Gool, L.V., Williams, C.K., Winn, J., Zisserman, A., 2010. The pascal visual object classes (voc) challenge. Int. J. Comput. Vis. 88, 303–338. https://doi.org/10.1007/s11263-009-0275-4.

Huang, L.K., Wang, M.J.J., 1995. Image thresholding by minimizing the measures of fuzziness. Pattern Recogn. 28, 41–51. https://doi.org/10.1016/0031-3203(94)E0043-K.

Jiang, N., Wang, J., Kong, L., Zhang, S., Dong, J., 2021. Optimization of Underwater Marker Detection Based on Yolov3, 187. Elsevier B.V., pp. 52–59. https://doi.org/10.1016/j.procs.2021.04.106

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. http://arxiv.org/abs/1412.6980.

Luhmann, T., 2014. Eccentricity in Images of Circular and Spherical Targets and its Impact to 3d Object Reconstruction. ISPRS. https://doi.org/10.5194/isprsarchives-XL-5-363-2014.

Luhmann, T., Robson, S., Kyle, S., Harley, I., 2006. Close Range Photogrammetry, 2. https://doi.org/10.1111/phor.12114. URL. http://trid.trb.org/view.aspx?id=814766.

Otsu, N., 1979. A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man, and Cybernetics 9, 62–66. https://doi.org/10.1109/TSMC.1979.4310076.

Padilla, R., Passos, W.L., Dias, T.L., Netto, S.L., Silva, E.A.D., 2021. A comparative analysis of object detection metrics with a companion open-source toolkit. Electronics (Switzerland) 10, 1–28. https://doi.org/10.3390/electronics10030279.

Paris, S., Kornprobst, P., Tumblin, J., Durand, F., 2008. A gentle introduction to bilateral filtering and its applications. ACM SIGGRAPH 2008 Classes. https://doi.org/10.1145/1401132.1401134.

Ren, S., He, K., Girshick, R.B., Sun, J., 2015. Faster {R-CNN:} towards real-time object detection with region proposal networks, CoRR abs/1506.0. http://arxiv.org/abs/1506.01497.

Reznicek, J., Luhmann, T., Jepping, C., 2016. Influence of raw image preprocessing and other selected processes on accuracy of close-range photogrammetric systems according to vdi 2634. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLI-B5 20. https://doi.org/10.5194/isprsarchives-XLI-B5-107-2016. https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLI-B5/107/2016/isprs-archives-XLI-B5-107-2016.pdf.

Sahoo, P.K., Soltani, S., Wong, A.K.C., 1988. A survey of thresholding techniques. Comput. Vis. Graph Image Process 41, 233–260. https://doi.org/10.1016/0734-189X(88)90022-9. URL. http://www.sciencedirect.com/science/article/pii/0734189X88900229.

Schneider, C.T., 1991. 3-d vermessung von oberflächen und bauteilen durch photogrammetrie und bildverarbeitung. Proceedings of the IDENT/VISION 91, 14–17.

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. http://arxiv.org/abs/1409.1556.

Wang, J., Olson, E., 2016. Apriltag 2: efficient and robust fiducial detection. IEEE International Conference on Intelligent Robots and Systems 2016-Novem 4193–4198. https://doi.org/10.1109/IROS.2016.7759617.

Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., Girshick, R., 2019. Detectron2.

Yen, J.-C., Chang, F.-J., Chang, S., 1995. A new criterion for automatic multilevel thresholding. IEEE Trans. Image Process. 4, 370–378. https://doi.org/10.1109/83.366472.

Zhang, Z., Hu, Y., Yu, G., Dai, J., 2022. Deeptag: a general framework for fiducial marker design and detection. IEEE Trans. Pattern Anal. Mach. Intell. X, 1–14. https://doi.org/10.1109/TPAMI.2022.3174603.