Synthetic Aperture Sonar Interferogram Filtering by Intensity Image Segmentation

Ole Jacob Lorentzen, *Member, IEEE*, Torstein Olsmo Sæbø, *Senior Member, IEEE*, Alan J. Hunter, *Senior Member, IEEE*, and Roy Edgar Hansen, *Senior Member, IEEE*

***Abstract*—Synthetic aperture sonar interferometry relies on the interferogram of two single look complex images to estimate bathymetry. The phase difference measurements have variance, which is typically reduced by spatial smoothing at the cost of horizontal resolution. The high resolution intensity image is related to the bathymetry because of the observation geometry. We therefore suggest an approach that constrains the filtering around edges found by intensity image segmentation. We demonstrate our suggested method on simulated data and show quantitative and qualitative improvements in both the horizontal resolution and the shape resolvability of small objects. We demonstrate a 30% improvement in RMSE of the bathymetric estimate, and observe that the estimated bathymetry more closely renders the real object shape for a small, but elevated object. We demonstrate our suggested method on real data and show similar results.**

***Index Terms*—Bathymetry, interferometry, mapping, sonar, synthetic aperture sonar (SAS).**

I. Introduction

SYNTHETIC aperture sonar (SAS) interferometry is a signal processing technique which builds on SAS to produce detailed, high resolution depth maps of large areas [1], [2], [3], [4]. The technique originates from its synthetic aperture radar counterpart [5], but there are different challenges, mainly from the slow speed of sound in water, which sets them apart in many areas [4]. While SAS has become the industry-leading state of the art tool for high resolution underwater imaging of larger areas, SAS interferometry for bathymetric depth mapping is only in recent years becoming commercially available. SAS bathymetric mapping introduces stricter requirements than systems that were focused solely on imaging. Creating high quality depth maps imposes requirements on the methodology and accuracy of both the bathymetric measurements, as well as the horizontal position estimates. With a high resolution system we are able to produce depth maps over large areas that render fine details of small objects.

SAS bathymetry relies on the phase difference between two vertically displaced sensors. This measurement has variance, and due to the speckle characteristic of rough surfaces when imaged with SAS [2], single samples may even tend towards full signal cancellation. It is therefore common to apply sample averaging to reduce the variance. This usually means to do spatial smoothing, at the expense of horizontal resolution. Therefore, we typically see a reduction of resolution on the scale of a factor ten in each horizontal dimension for bathymetry, compared to the intensity image [3].

In this article, we consider the state of the art SAS system HISAS mounted on a HUGIN AUV, both developed by Kongsberg Discovery and the Norwegian Defence Research Establishment (FFI). The resolution of this system is about 2 cm horizontally, which is reduced to about 20 cm for the bathymetry due to required spatial averaging.

Mapping and detecting small objects on the seafloor with SAS is a field with many applications in geological [6] and archeological [7] sciences, as well as in military contexts, in particular for mine target detection [8]. Being able to utilize bathymetric mapping as well as the intensity image is a huge benefit, because the derived geometry is independent of observation angle, as opposed to the sonar image which is a 2-D projection of an object [9]. With traditional sonar detection and classification, the 2-D projections of 3-D objects may limit performance [10], [11].

In this article, we suggest a method to improve the effective resolution in SAS bathymetry, and in particular achieve better conservation of target shapes. We utilize the high resolution intensity image to guide the inclusion of samples for averaging the interferogram to produce the bathymetry estimates. The study builds on the preliminary results we published in [12]. We find that our method achieves more accurate object rendering and better separation ability for closely located objects, with lower variance of the estimated bathymetry in comparison to fixed window filtering.

Interferogram filtering in SAS has been studied extensively, but even more so in the SAR community where they face the same challenges. Filter classes range from the very common and robust local mean boxcar filter, to nonlocal filters and deep learning methods. Local filters with fixed window sizes will often break stationarity assumptions, as well as reduce the resolution. The more advanced methods represent various ways

Manuscript received 30 March 2023; revised 19 January 2024 and 28 February 2024; accepted 3 March 2024. This work was supported by FFI. *(Corresponding author: Ole Jacob Lorentzen.)*

Associate Editor: M. Hayes.

Ole Jacob Lorentzen is with the University of Oslo, 0313 Oslo, Norway, and also with the Norwegian Defence Research Establishment (FFI), 2007 Kjeller, Norway (e-mail: ole.lorentzen@ffi.no).

Torstein Olsmo Sæbø is with Norwegian Defence Research Establishment (FFI), 2007 Kjeller, Norway.

Alan J. Hunter is with the University of Bath, BA2 7AY Bath, U.K., and also with the University of Oslo, 0313 Oslo, Norway.

Roy Edgar Hansen is with the University of Oslo, 0313 Oslo, Norway.

Digital Object Identifier 10.1109/JOE.2024.3374465

to mitigate this issue. Xu et al. [13], provided a helpful overview and organizes the filters as follows.

1) Local: Filtering by local windows or regions.
2) Transformed-domain: Filtering by exploiting frequency and wavelet transformations.
3) Nonlocal: Filtering by evaluating patches for global similarity, not restricting the search locally.
4) Deep learning: Filtering using deep learning algorithms, including sparse methods and others.

Local filters applies the filtering operation by windows or regions locally. Examples include sliding window filters, Lee filters [14], varying window shapes and sizes, local-frequency estimator filters, and hybrid methods attempting to separate large scale and fine-scale variations. A very common example is the *boxcar* filter, which is simply a fixed window moving average. It is the maximum likelihood interferometric phase estimator if the phase is locally stationary and the scene reflectivity is locally homogeneous [15]. While this filter is very stable and provides predictable results, we find that these assumptions often break due to bathymetry variations and strong intensity variations, both of which are typically observed on objects and features containing sharp edges, as well as with man-made objects. This is the motivation for applying more exotic filtering to the interferogram estimation process.

Transformed-domain filters exploit frequency and wavelet transformations. In this group we find the Goldstein filter [16], which is one of the most popular filters in the SAR community. It filters the interferogram by a kernel derived from the smoothed intensity of the 2-D FFT spectrum of the interferogram itself. This should provide heavy smoothing in homogenous areas where the stationarity assumption is valid, while preserving high frequency changes in the interferogram values in other regions. It has been very successful with SAR data, because of good performance and ease of implementation. However, it can in many cases also produce filtering artifacts in the resulting interferogram, even when using weighted, overlapping estimation windows as described in Goldstein's original paper. Many modified Goldstein filters have been proposed, typically using adaptive values for the filtering strength parameter derived from coherence, phase standard deviation or image intensity [17], but also modifying the filtering process entirely, which has shown promising results [18], [19], [20].

Nonlocal filters and deep learning methods have shown impressive results, but are typically sensitive to the algorithm parameters and input data conformity with respect to the model or training data [13]. We will therefore not consider methods from these classes in this study.

While we can generalize that the potential accuracy of these methods increases from the local filters through to the deep learning methods, so does the method and implementation complexity, as well as the sensitivity to algorithm parameters [13]. Therefore, from a robustness standpoint, it is reasonable to consider the local filters for general processing, rather than the more complex variants.

The method we propose in this study falls in the class of local filters. Briefly outlined, we apply an adaptive filter shape to the interferogram, which is not restricted to predefined window shapes as in boxcar or Lee filters, but allows each local filter shape to vary independently, based on the intensity image values. This can be related to adaptive neighborhood region-growing techniques such as the ones described in Vasile et al. [21] and Song et al. [18], but driven by an independent, intensity image segmentation instead. Vasile et al. [21] argued that samples with similar intensity were more likely to respect the stationarity hypothesis, which was also in agreement with the assumption of homogeneous reflectivity. While applied to polarimetric SAR images, their method resembled our proposed method, with a significant difference in that we have separated the intensity image segmentation as an isolated problem. Our approach thus allows the application of an arbitrary segmentation algorithm without interfering with the interferogram estimation part of the method. Using image segmentation for this purpose potentially allows better object shape resolvability, as well as less variance in homogenous regions by allowing the use of larger window sizes without compromising the resolution near objects.

The rest of this article is organized as follows. First, we describe the SAS signal processing, image segmentation and the interferogram estimation in Section II. In Sections III-A, III-B, III-C, III-D, and III-E, we apply our method to a simulated scene and compare the results to other methods and the ground truth. In Sections III-F, III-G, and III-H, we apply our method to a real data example and compare the results to other methods and the available ground truth scales. Finally, Section IV concludes this article.

II. Method

We produced the single-pass interferogram from two SAS images collected from two receiver arrays displaced by a vertical baseline. Then, we performed image segmentation on the intensity image from the master bank, providing us with a segment map. In the interferogram filtering step, we used this segment map to restrict the sample averaging only within a segment. Finally, we produced the depth map from the filtered interferogram. In the following sections, we describe the processing steps in greater detail.

A. Imaging and Interferometry

We produced high quality single-look complex SAS images by a back-propagation algorithm [22], [23] on flat rendering planes at the estimated mean depth. We used processing grid resolution of 2 cm, slightly oversampled from the theoretical spatial resolution. We denoted the images $I_{\text{upper}}(x, y)$ and $I_{\text{lower}}(x, y)$, where x and y represents the image pixel coordinates, and the subscripts indicate which receiver array (bank) is used. We then produced the interferogram [3], [5] by calculating the phase difference between the vertically displaced receiver banks as

$$u(x, y) = I_{\text{upper}}(x, y) I^{*}_{\text{lower}}(x, y) \tag{1}$$

where the * indicates the complex conjugate. We denote the interferometric phase as

$$\phi(x, y) = \angle u(x, y). \tag{2}$$

The lower bank SAS image is designated as the master image because it typically contains less noise than the upper image as a result of being furthest away from both acoustic communication payloads and the sea surface in shallow water operations. The master image was used for the image segmentation. Interferogram averaging was performed over a moving square window. In our suggested method, the moving square window is modified to not include values across segment boundaries, as described in Section II-C. The resulting interferogram is then used to produce the bathymetric map [3]. Finally, these depth estimates are used to correct for geometric errors in the a priori imaging plane.

B. SAS Image Segmentation

SAS backscatter image segmentation is a field with close relation to SAR image segmentation, and could be achieved with numerous algorithms [24]. In this article, we have opted to use a simple segmentation algorithm in order to illustrate our concept of interferogram filtering. Our segmentation algorithm attempts to separate segments of strong backscatter intensity from background, consisting of lower intensities, speckle, and shadow. Finding an optimal segmentation algorithm is outside of the scope of this work. Instead, we focus on the problem of selecting the correct samples for the interferogram averaging.

We first preprocessed the image, then performed a speckle-reduction step, before applying k-means segmentation [25]. This provided us with a segment map. Finally, we postprocessed the segment map in order to avoid very small segments, as we required some sample averaging to take place.

To preprocess the image, we first limited the SAS image dynamic range from the maximum intensity value to provide a higher contrast image. Then, we normalized the data values in dB to a [0,1] interval to conform with image processing norms. We reduced speckle variability through filtering by an edge-preserving smoothing with the nonlocal means (NLM) algorithm [26] using a degree of smoothing of 1σ. To further reduce noise in the filtered image, we then filtered by performing a simple dilation and erosion filtering on the image using a filter size of 3×3 samples. After applying k-means segmentation, we postprocessed the segment map by applying a threshold on the number of samples in each segment.

We chose the parameters for these filtering and segmentation steps empirically. In Section III-D, we show results from varying them. They can be further tuned or determined differently in future work. From experience, we have about 50 dB useful dynamic range in images, and we decided to use 30 dB dynamic range from the maximum value to increase the image contrast and highlight strong targets. In the NLM-filtering, we used a degree of smoothing of 1σ, where σ is the standard deviation estimated from the input samples. For the following filtering by dilation and erosion, we used the smallest, symmetrical filter size of 3×3 pixels. We found that it left us with significantly less chance to have residual high frequency noise in the filtered images at this point. The number of segments in the k-means segmentation is a sensitive parameter which must be chosen carefully. We intend to separate objects (highlights), from background, typically dominated by seafloor and shadows. We therefore used $k \in [2, 4]$. We found that in practice we often got better performance with fewer segments in the simulations, while the real data example performed better with more segments. This is because increasing the number of segments did not have the intended effect of separating highlights, shadows, and speckle seafloor, but rather produced halos around objects and *bridges* between segments in the simulation. Other algorithms could potentially handle this better, and there are also specific algorithms for estimating the number of segments to use for a given image before running k-means segmentation. However, a study on segmentation techniques is outside the scope of this work. We used the described approach to simplify the segmentation problem to something that works well enough to study the proposed interferometric filtering technique, and demonstrate the performance improvement. With the only requirement being to take a single-look complex SAS image as input and providing a segment map as output, the segmentation algorithm could be interchanged seamlessly in this processing chain with any desired algorithm.

The result of this image processing is a segmented image mask $M(x, y)$ indexed by an integer class number. For each pixel, its segment was identified by a connected components algorithm which considered segments to be continuous if the edges or corners between two pixels belonging to the same segmentation class touch, i.e., so that the pixels that are connected to (x, y) are

$$(x \pm 1, y) \text{ or } (x, y \pm 1) \text{ or } (x \pm 1, y \pm 1). \tag{3}$$

Even after the described filtering steps, we sometimes had a few very small segments consisting of only a few samples. These were related to speckle noise realizations. To mitigate these, we applied a continuous segment threshold when filtering the interferogram, asserting more than five samples in any continuous segment. If this requirement was not met, the segment was removed and the previous sample's segment applied, which would typically be the seafloor segment.

Finally, we had a set $\xi(x, y)$ per pixel of included pixels that are both within the original filtering window and in the same segment.

C. Segmented Window for Interferogram Averaging

Square window interferogram filtering is described as

$$v(x, y) = \frac{1}{N} \sum_{(i,j) \in \psi(x,y)} u(i, j) \tag{4}$$

where $\psi(x, y)$ is a set of the pixels belonging to the square window around pixel (x, y), and N is the number of pixels in the window. With the segments found by the method described in Section II-B, we modified (4) to only include pixels inside the rectangular smoothing window that were in the same segment as the current, center pixel. We then obtained the filtered interferogram $w(x, y)$ after averaging with segment preservation by

$$w(x, y) = \frac{1}{N(x, y)} \sum_{(i,j) \in \xi(x,y)} u(i, j) \tag{5}$$

where $\xi(x, y)$ is a set of the pixels belonging to the segmented window around (x, y) as described in Section II-B, and $N(x, y)$

represents the number of pixels included for each set. This eliminated averaging across segment boundaries.

III. Results

To analyze the results, we compared our method to fixed square window averaging in the interferometric processing. In simulations, we also compared the results to bathymetry processed with ground truth input and an ideal interferogram produced by averaging 81 realizations with independent scatterer positions and no spatial smoothing, corresponding to a window size of 9×9 samples.

We assessed the results by comparing the interferometric coherence, and by visual inspection of the estimated bathymetry and the bathymetry-corrected backscatter image. On simulations, we also compared the results with the ground truth bathymetry and calculated root mean squared errors (RMSE). On the real data set, we also analyzed the result in regard of our available knowledge of the target geometry.

A. Simulator and Scene

We used a sonar point scatterer simulator to produce raw time series of complex hydrophone data from a single pass of a flat, 5×5 m seafloor with an arrangement of eleven cylinders of varying heights and diameters. We used 16 scatterers on average per 2×2 cm resolution cell to produce fully developed speckle. We simulated the HISAS SAS transducers on an ideal, straight-line trajectory and ideal environmental conditions, with additive, white noise of 20-dB signal to noise ratio (SNR), measured relative to the speckle seafloor. The measured intensity of the targets are about 17 dB stronger than the surrounding seafloor. The sonar is located at depth $z = 0$, resulting in a grazing angle of about 35°. This results in shorter shadows, but longer layover regions compared to a smaller grazing angle, i.e., targets at longer range. We use a geometry-based shadow calculation per ping to omit occluded scatterers from the simulation. We did not include multipath scattering in the simulation. We used the same SAS processing chain described in Section II on the simulated raw data as for real data in Section III-G.

We arranged eleven cylinders in a square pattern with varying distances between them to facilitate measuring the resolution capabilities of the algorithms in both the along- and cross-track directions. We required nonsquare targets to study the shape resolvability of our method compared to square window filtering. In Fig. 1, we show one realization of the simulation, which provides a reference of the scene geometry. All the cylinders are 20 cm in diameter, except for the two smaller ones in the middle of the right-most, vertical line of cylinders. They have diameters of 5 and 10 cm, respectively, from the top. All the cylinders are also populated with scatterers on the sides. The heights from the seafloor ranges from 10 to 25 cm in 5 cm intervals. The gaps between the cylinders are 10, 20, and 40 cm. These dimensions and spacings were chosen to make the targets large enough to roughly fit a full square 9×9 sample window inside each cylinder, while having spacings ranging from values larger than the expected 20-cm resolution of square window filtering and down to spacings only resolvable in the image resolution. The smaller cylinders are designed to test the minimum object size, with the smallest one representing a target with only about four samples. This is expected to be smaller than the design resolution for our method since we threshold the segments at minimum five samples. The heights of the cylinders were varied to give a verification of the depth estimates and easily separate them from each other.

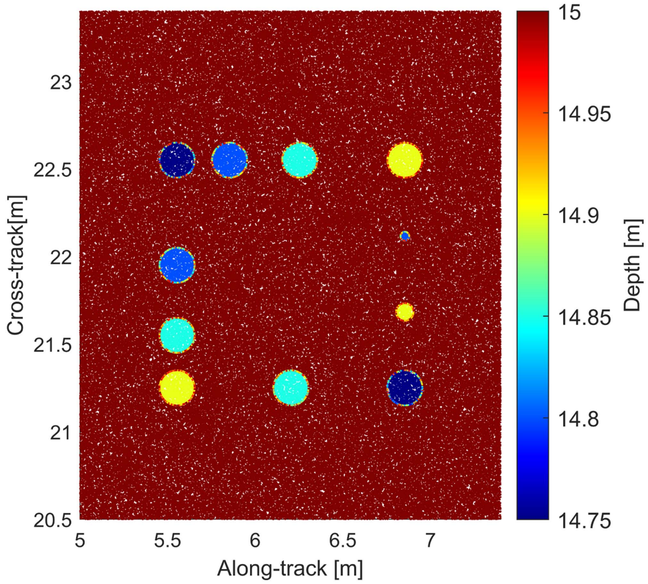


Fig. 1. Realization of simulated scatterers illustrating the scene geometry. The noisy halo around each object is the rendering of random scatterers positioned vertically on the sides of the cylinders.

For the special case of multiple realizations that we describe in Section III-C, we matched the 9×9 window size and generated 81 different, random realizations of the scatterer positions and reran the full simulation for each of them.

B. Simulation Results

In Figs. 2 and 3, we show a SAS image of the simulated scene and its segmentation using two segments. Notice that the smallest cylinder is detected as a single pixel, but ignored by the filtering which removes tiny segments, as described in Section II-B. The lower-right cylinder partially overshadows the 10 cm cylinder after it in range, which results in it not being properly visible in the image, and thus not detected by the segmentation algorithm either. The other cylinders appear to be identified fairly well by the segmentation algorithm.

In Fig. 4, we show the number of samples used in the estimated interferogram when using segmentation filtering. Notice how the number of pixels defaults to the full 9×9 samples square window when not near a segment edge, and reduces to about half of that when at an edge. We also observe the edge effect when parts of the filter move outside of the data area, which was present for all the methods. We will not be considering edge effects in this study, as they can easily be avoided by proper overlap and cropping in the higher level processing chain.

In Fig. 5(a), we show the bathymetry estimate using square window filtering, and in Fig. 5(b) the corresponding bathymetry

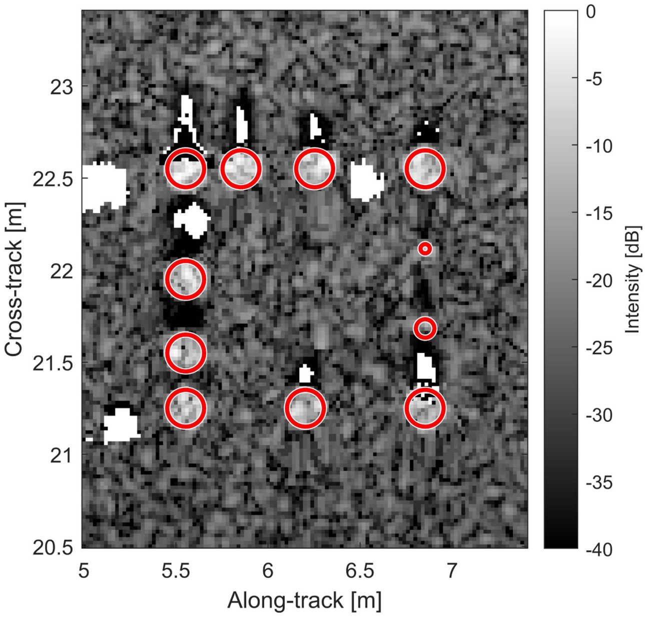


Fig. 2. Simulated SAS image with an array of cylinders.

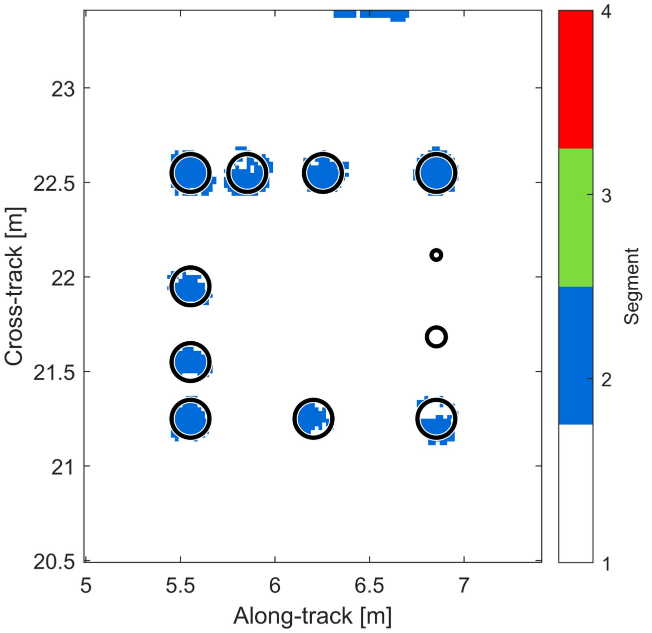


Fig. 3. Simulated SAS image segmentation map with two segments.

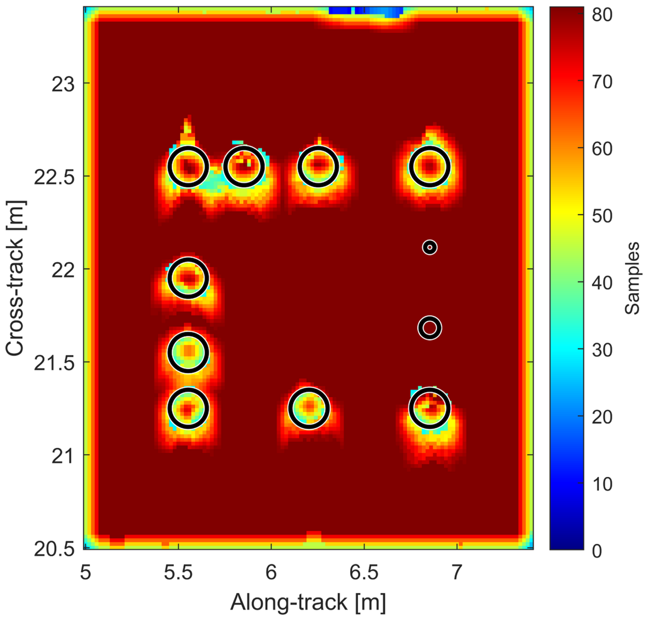


Fig. 4. Number of samples per depth estimate for the simulated scene using segmentation filter with two segments.

estimate using image segmentation filtering. The ground truth outlines of the cylinder objects are plotted as solid, black circles.

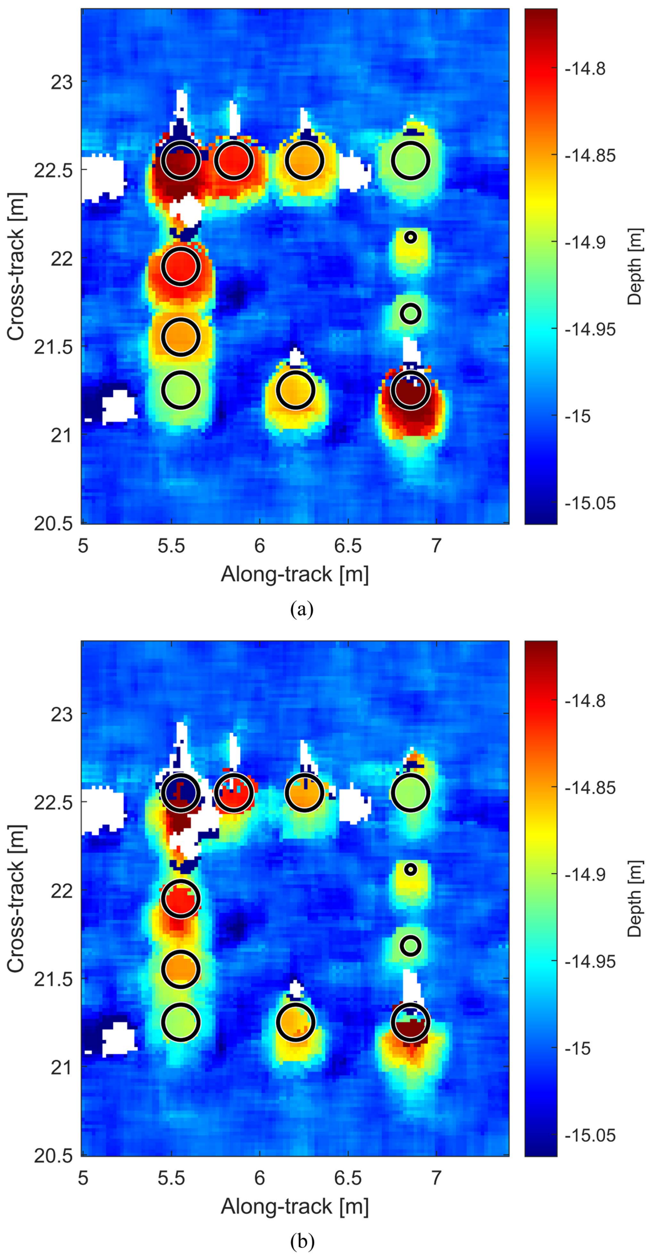
To quantitatively compare the resolution between the various methods, we produced a slice through the objects as we illustrate by the white, solid lines in Fig. 6. In addition, we have selected a 10 pixel slice that covers the cylinder objects, illustrated by a translucent, white overlay in Fig. 6. We calculated the depth RMSE for both the 1 and 10 pixel slices. We chose these slices to not include the surrounding seafloor, as it is not relevant for studying this method. The flat seafloor area where no segments were generated in Fig. 3 will fall back to the same method (full square window) and be numerically identical. By restricting the slices to the width of the cylinder targets, we were able to separate issues of shadow and layover to be primarily present only in the cross-track case. This enabled us to study the resolution without regard to shadow and layover in the along-track case.

In Figs. 7 and 8, we show the various bathymetry estimates for the slices along-track and cross-track. In Fig. 9, we plot the average RMSE values for each method, separately for the along- and cross-track directions. In Fig. 9(a), we show the RMSE for the single pixel slices, and correspondingly the errors for the wide slices are in Fig. 9(b). For comparison, we calculated corresponding slices in the flat region and got a depth RMSE value of 0.5 cm using a 9 × 9 window. With the smaller 3 × 3 window we observed the large increase in variance with depth RMSE of 1.2 cm, while the larger 17 × 17 window achieves about 0.3 cm. These values were the same for both methods, as our method falls back to the square window in these regions.

C. Simulation Processing With Multiple Speckle Realizations and Ground Truth Input

In order to create an ideal reference bathymetry estimate, we chose two different approaches. The motivation for both was to indicate what the best achievable results are with the applied SAS processing chain, if we had ground truth a priori knowledge of the depth, or idealized measurements.

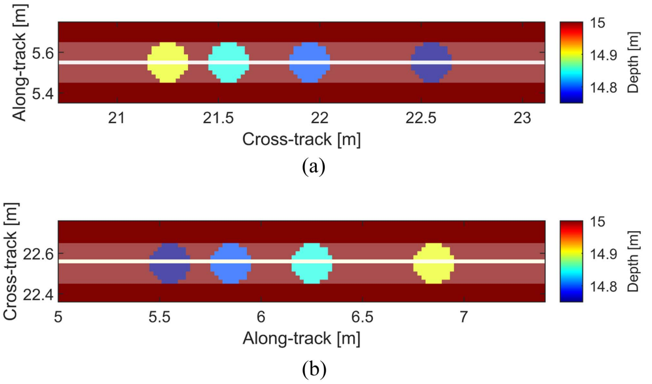
For the first approach, we created the SAS images using the processing chain as described before, but with the ground truth bathymetry as render plane for the images, which also followed through for the interferometric processing. This reduced the task for the interferometric processing to that of estimating a zero-valued phase difference, since we would obtain the ground truth result if all the phase angles are zero. We applied the same interferogram filtering with a 9 × 9 square window and image feature segmentation with two segments.



(a)

(b)

Fig. 5. Estimated bathymetries projected in ground plane. The circles outline the ground truth cylinder positions. (a) Square 9 × 9. (b) Segmentation, two segments.



(a)

(b)

Fig. 6. Ground truth bathymetry with single and 10 pixel slices indicated by white overlays. Notice the changing axes directions. (a) Cross-track slice. (b) Along-track slice.

For the second approach, we created 81 independent realizations of the simulation. The scatterer positions were independent and random, and thus so was the speckle in the scene. The additive noise was also independent and random for each realization. We averaged all the realizations pixel-wise instead of using any spatial filtering, resembling an ideal 81-look interferogram. The 81 independent samples per pixel were comparable to a 9×9 square window filtering, but as a reference result in which the real expected value of all samples were correct with no spatial smoothing. This should be an approximation of the best possible resolution that could be obtained before the imaging system parameters are the limiting factor. While useful as a hypothetical case for a simulation, producing such a data set in real life is not feasible. It would require 81 exactly repeated passes with changing the seafloor speckle in between each pass, e.g., by moving particles on the seafloor while simultaneously not perturbating the actual depth.

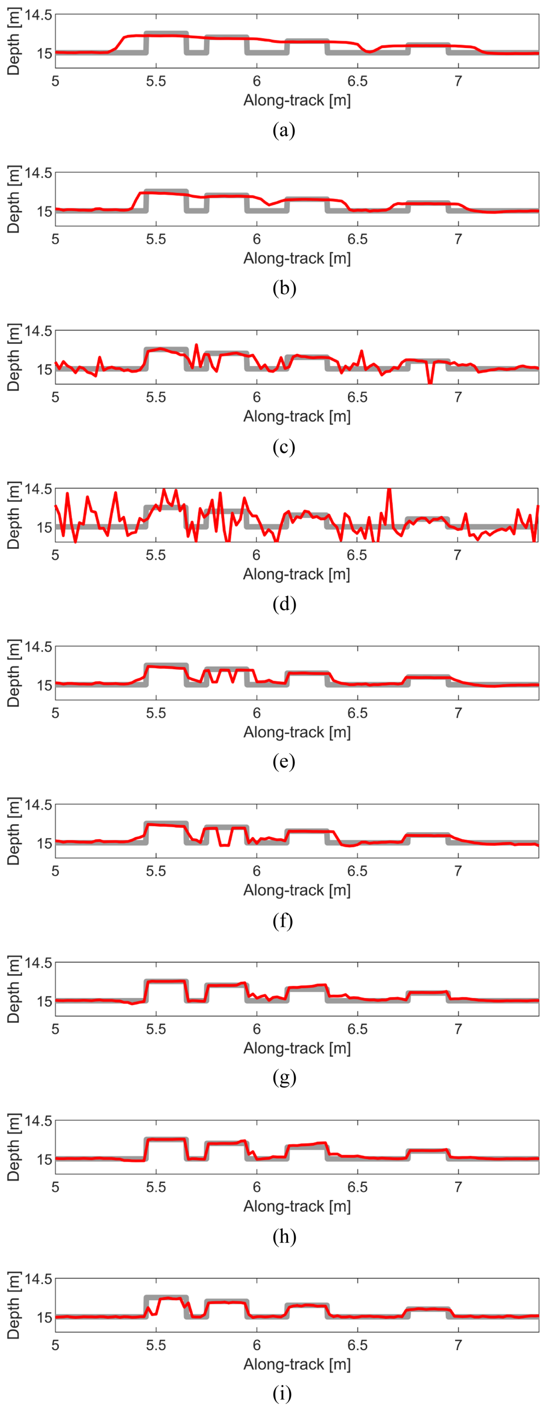
D. Segmentation Algorithm Parameters

We have evaluated the segmentation algorithm by varying the available parameters. In Fig. 10, we show aggregated results by calculating the RMSE values comparable to those presented in Fig. 9. In this section, we describe the results for each parameter in detail.

The dynamic range of the input image significantly changes its appearance and thus the performance of further image processing. We expect a useful dynamic range up to 40–50 dB with our system, on which we have based the simulation. In Fig. 11, we show three examples of varying dynamic ranges. At 10 dB we barely see the target highlights. At 50 dB, we observe significant noise in the shadow regions. We expect our method to perform best in-between these extremes, and in practice we observed very similar segmentation performance from 30–50 dB dynamic range. In Fig. 10, it is clear that the 50-dB dynamic range performs better than 30 dB on this simulation. However, with real data examples we observed that 50 dB in many cases introduced too much noise to achieve a good segmentation result. We therefore decided to maintain a lower dynamic range for the simulation example for consistency.

The degree of smoothing in the NLM-filtering is given as a scaling of the standard deviation, which is estimated from the intensity image itself. This parameter controls the window size in the NLM-algorithm, and is 1σ in the original paper [26]. Scaling it to half for a smaller window size seems to perform fairly well in Fig. 10, but the resulting segmentation map appears noisy and with uneven segment edges upon inspection. Increasing the window size by scaling up this parameter appears to increase the along-track error, so we find 1σ to work best.

The window size for the dilation and erotion filters were evaluated at 3×3 (*Feature* 9×9, *three segments*) and 5×5. Neither stand out as a significantly better choice from the error values in Fig. 10. However, increasing this window size also



(a)

(b)

(c)

(d)

(e)

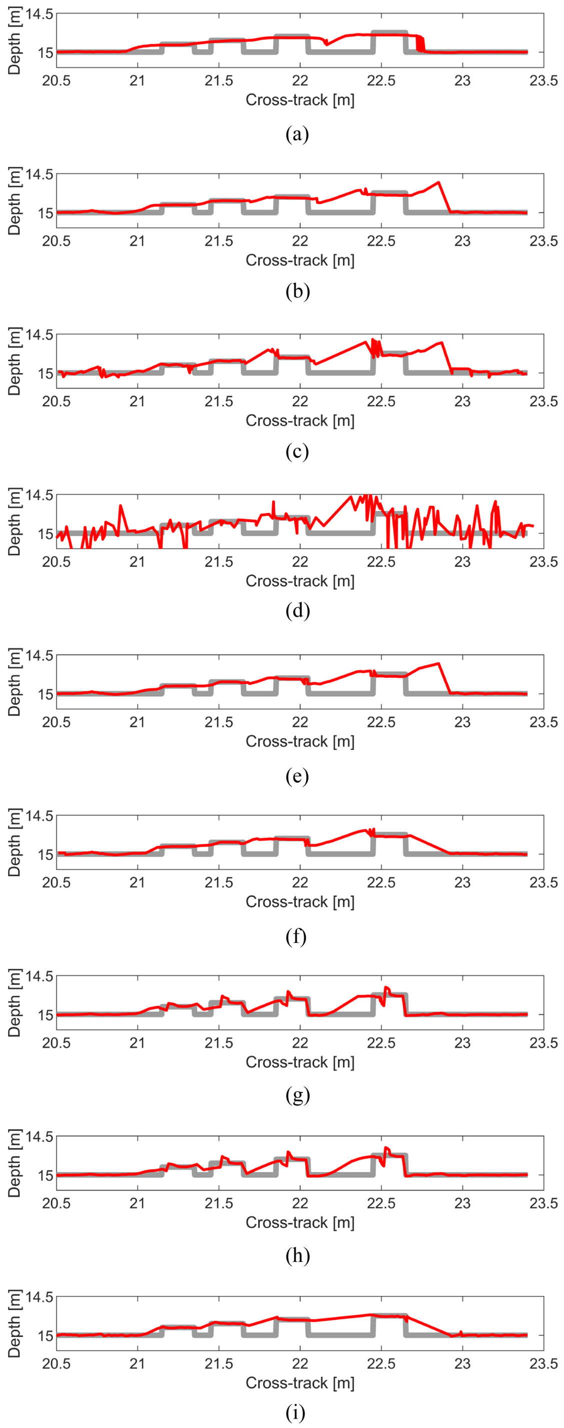
(f)

(g)

(h)

(i)

Fig. 7. Single-pixel bathymetry slices across the objects along-track for various processing options. The gray lines represent the ground truth bathymetry. (a) Square 17×17. (b) Square 9x9. (c) Square 3×3. (d) Square 1×1. (e) Segmentation, two segments. (f) Segmentation, three segments. (g) Square 9×9, ground truth rendered. (h) Segmentation, two segments, ground truth rendered. (i) 81 speckle realizations.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

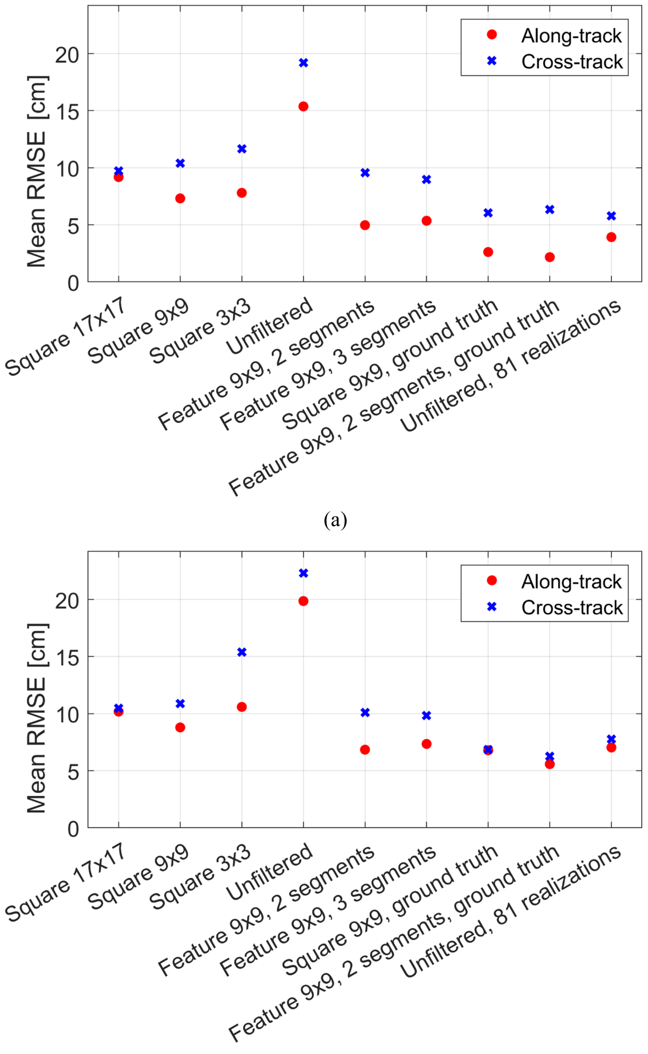
(h)

(i)

Fig. 8. Single-pixel bathymetry slices across the objects cross-track for various processing options. The gray lines represent the ground truth bathymetry. (a) Square 17×17. (b) Square 9×9. (c) Square 3×3. (d) Square 1×1. (e) Segmentation, two segments. (f) Segmentation, three segments. (g) Square 9×9, ground truth rendered. (h) Segmentation, two segments, ground truth rendered. (i) 81 speckle realizations.

limits our resolution ability. Therefore, we have chosen to keep this at the lower size.

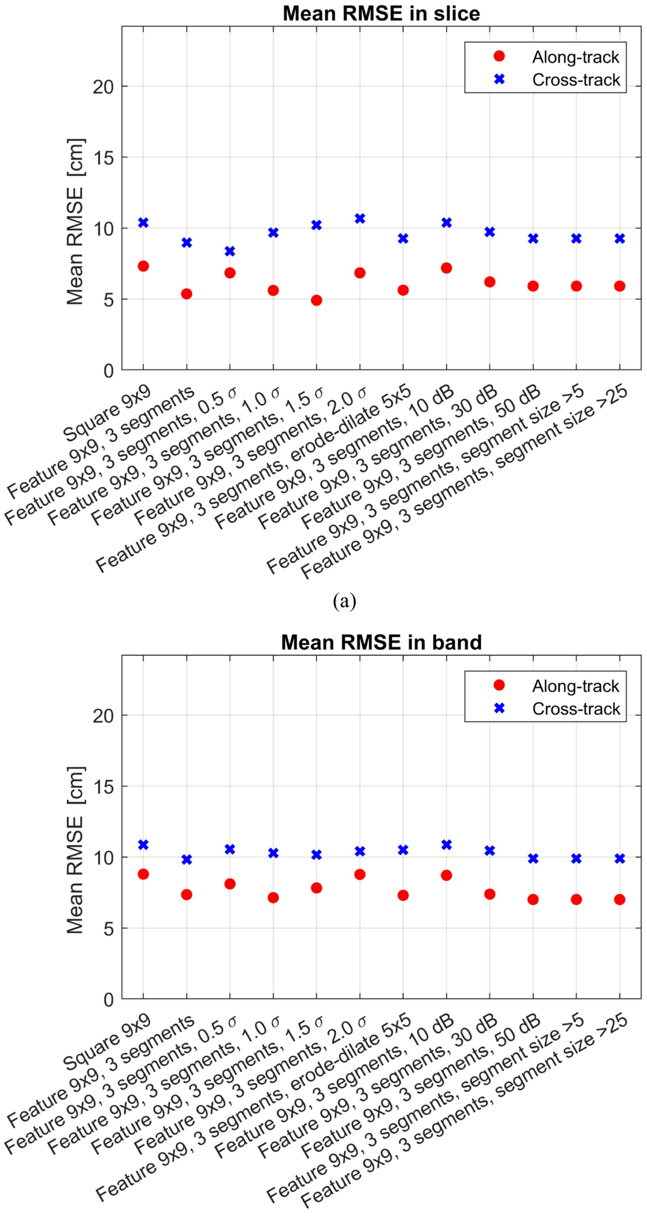
The number of samples required for the continuous segment threshold was set to five samples. This was not a significant parameter in the simulation, and upon inspection it does not remove any segments in this case. Increasing it to 25 samples removes a few segments, but does not seem to impact the errors in Fig. 10 significantly. Again, this also limits our resolution ability, which is even observable by the error values in Fig. 10(a) which increase slightly compared to the band result in Fig. 10(b).



(a)

(b)

Fig. 9. Root mean squared errors in centimeters for various methods in along-track and cross-track directions. (a) Errors for the single pixel slices. (b) Errors for the wider slices covering the cylinder objects.



(a)

(b)

Fig. 10. Root mean squared errors in centimeters for various segmentation algorithm parameters in along-track and cross-track directions. This figure can be directly compared to Fig. 9. (a) Errors for the single pixel slices. (b) Errors for the wider slices covering the cylinder objects.

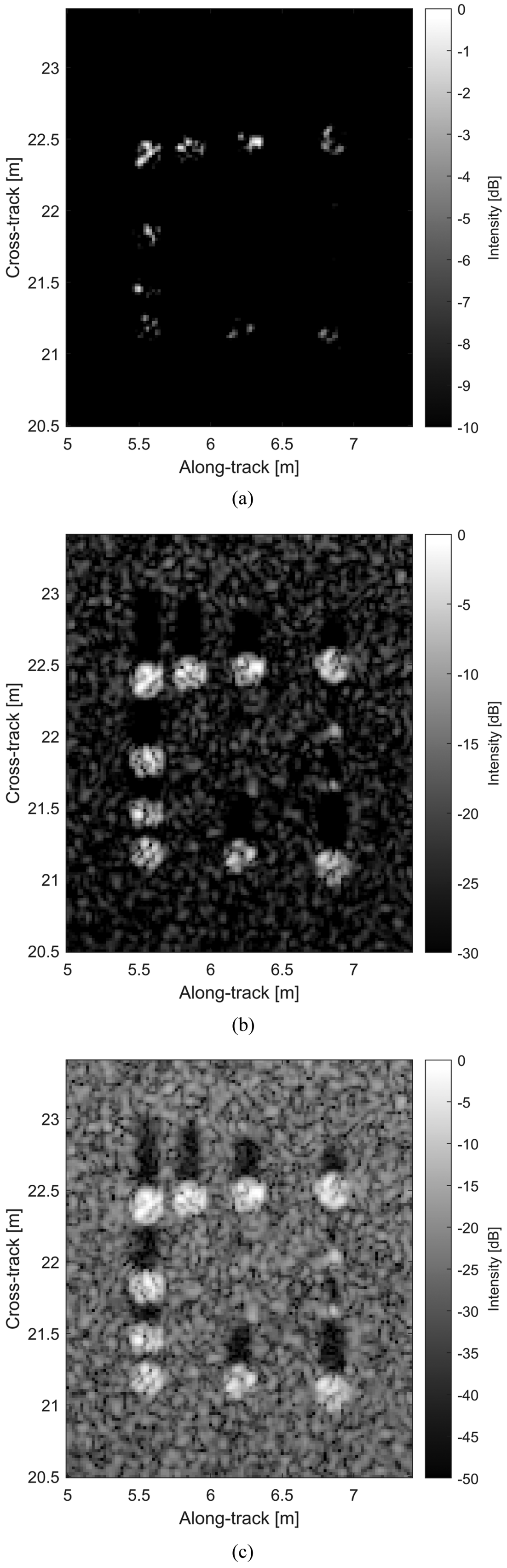
E. Simulation Discussion

Comparing the plots in Fig. 5, we observe that both methods estimate the depth of each object well, but with our method we find that the boundaries of the depth estimates are closer to the ground truth outlines. We also observe that the seafloor in-between the cylinders was better resolved with our method. This represents improved shape resolvability and improved horizontal resolution, respectively.

In the aggregated results presented by Fig. 9, we observe that our method gives only a slight improvement in the cross-track direction. This is due to shadow and layover. In the along-track direction, our image segmentation method gave significant improvements compared to the corresponding square window methods. For the wide slices we approached the performance of the reference methods, which utilized ground truth bathymetry input. In all the cases studied here, our suggested method outperformed the traditional fixed-window approach.

For both the ground truth input approaches, we find the results plotted in comparison to the other methods in Figs. 7 and 8, and more concisely in Fig. 9. We observe that they generally perform the best as expected, and that our method gives estimates closer to these ground truth methods. We find a significant improvement in RMSE of about 30% for the along-track case. For the cross-track case we find only minor improvements, because this is limited by shadow and layover.

We evaluated the segmentation algorithm by varying the available parameters, and found that the performance gain of our algorithm is present regardless of the parameter settings. The performance varies with the parameters, but appears to fall back



(a)

(b)

(c)

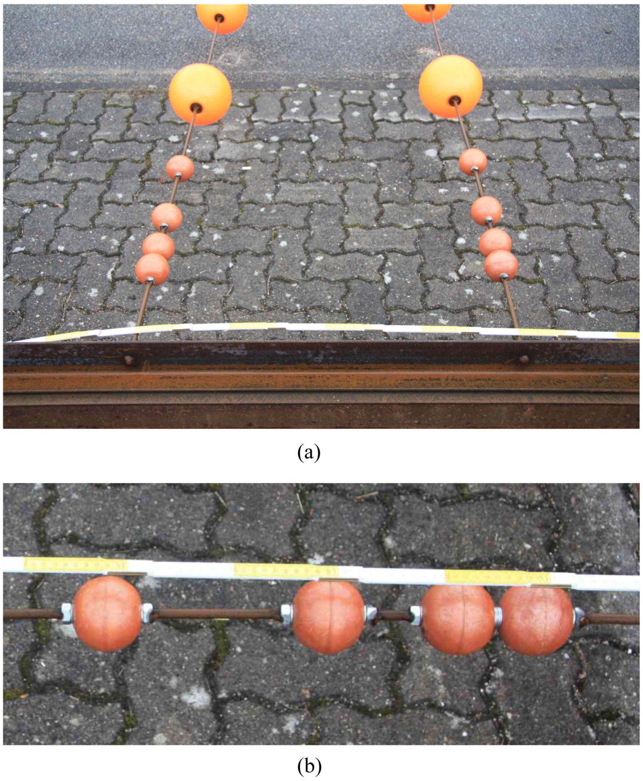
Fig. 11. Simulated SAS intensity images displayed with different dynamic ranges. (a) Dynamic range of 10 dB. (b) Dynamic range of 30 dB. (c) Dynamic range of 50 dB.

to the square window performance as expected. The cross-track performance appears to be better than along-track regardless of parameters.

As mentioned earlier, our method falls back to the square window method when no segmentation boundaries are present within the window. Increasing the window size with our suggested method is reasonable because we are less prone to smoothing out features, and a larger window size where applicable is desirable to further reduce the variance. Our method could be extended to adapt the window size within the current segment so that we maintain a constant number of samples per pixel, with the benefit that the resulting estimate statistics would be more homogenous. A different strategy would be to increase the window size significantly, maybe with the inclusion of a distance weighting filter. Properly tuned we would expect this to have similar characteristics and performance as an adaptive window, but the implementation might be technically easier. Extending the method for this particular detail is outside of the scope of this study. We have chosen not to implement this because we found that we got sufficient results without taking such measures. This is in part due to the averaging process already being a weighted average, in which the high intensity samples contribute more to the average phase estimate. This is what caused the significant dilution of the object bathymetries in the first place. However, since the objects which causes segment boundaries typically have high intensity, we will also have strong signal and thus good phase estimates in these cases, even if we have fewer samples in the averaging process. The opposite was the case for the adjacent seafloor samples, which could clearly have benefitted from an adaptive window size as discussed here. However, the elimination of the strong object scatterers enabled us to get more accurate phase estimates for these cases as well, even with fewer samples.

F. Real Data Set

We used SAS data from the HISAS interferometric synthetic aperture sonar on a HUGIN AUV. The data was collected by the Bundeswehr Technical Center for Ships and Naval Weapons, Maritime Technology and Research (WTD 71) in 2010 near Eckernförde, Germany. It contained a deployed frame with various resolution targets, of which we investigated arrays of spheres with various spacings. In the upper panel of Fig. 12, we show an overview of two of the sphere arrays. In each array we had four closely spaced small spheres with 7 cm diameters, and some larger spheres with larger separation. We focused our study on the small spheres because ground truth measurements were available for these. Notice also that there was a small bolt at the end of the center pole to fix it, which also showed up in the SAS image as a significant highlight. In the lower panel of Fig. 12, we show a closer view of the small spheres with a ruler for scale. The spheres were laid out with center to center distances of about 21, 13, and 8 cm, giving spacing between the spheres as 14, 6, and 1 cm. These distances were suitable for our study because they are between the image and bathymetry resolutions. Imaging resolution with this system is slightly larger than 2 cm, which should make us able to resolve all


(a)

(b)

Fig. 12. Sphere array in the resolution frame target. (a) An overview of two of the five sphere arrays. (b) A close-up of the small spheres under investigation.

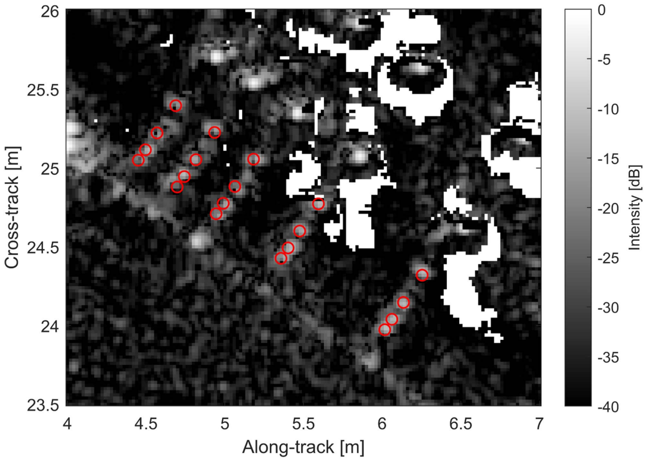


Fig. 13. SAS image of the array of spheres. The red circles are target outlines manually superimposed on top of the images as a guide, but do not represent known ground truth positions of the targets.

but the smallest distanced spheres. The bathymetric resolution is however typically reduced to nine times that, resulting in 18 cm horizontal resolution in the bathymetry. Hence, we would not expect to be able to resolve any of these spheres in the bathymetry without very small estimation windows or adaptive processing.

G. Real Data Results

In Fig. 13, we show the SAS image in the ground plane. In Fig. 14, we show the corresponding segmentation using four segments. The circles superimposed on the images are outlines of the sphere positions. These were drawn with correct dimensions and distances, but they were placed manually to act as a guide and do not represent ground truth position of the spheres.

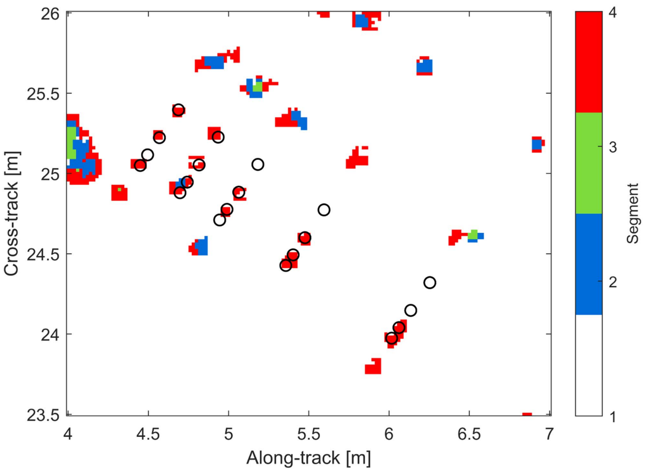


Fig. 14. Example of a SAS image segmentation map of the array of spheres. The black circles are target outlines manually superimposed on top of the images as a guide, but do not represent known ground truth positions of the targets.

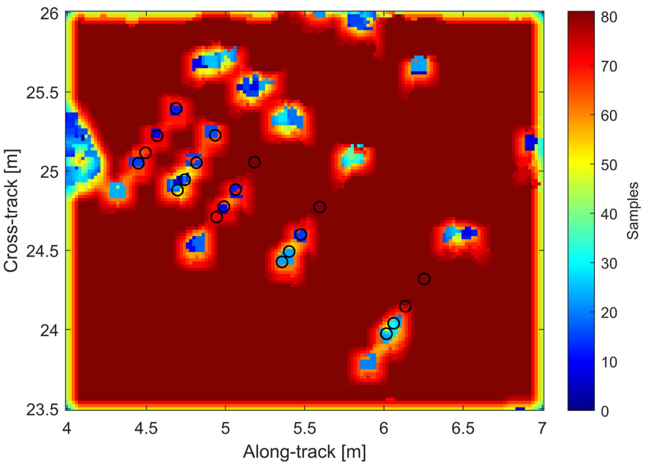
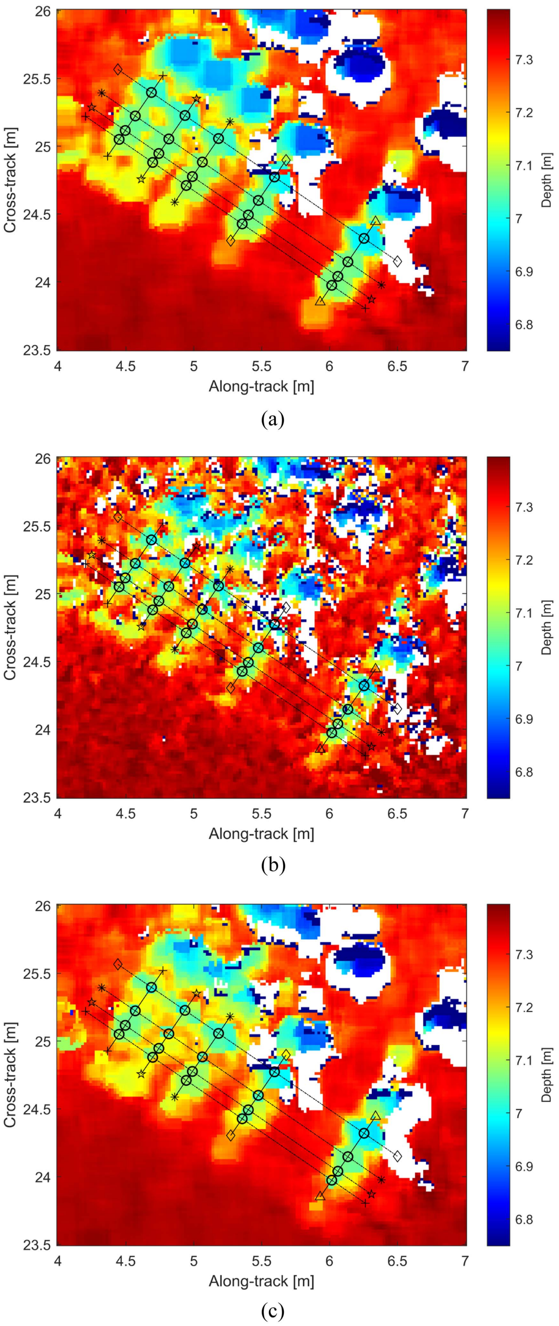


Fig. 15. Number of samples per depth estimate for the real scene shown in Fig. 14 using segmentation filter with four segments. The black circles are target outlines manually superimposed on top of the images as a guide, but do not represent ground truth positions of the targets.

In Fig. 15, we show the number of samples used in the estimated interferogram when using segmentation filtering. Notice that the number of samples drops significantly with the small sphere segments, in which case we are only using 10–30 samples. We observed an increased interferometric coherence on the targets caused by the exclusion of the surrounding samples, which were not coherent. The seafloor surrounding the targets was also affected with reduced number of samples, and here we observed a lower interferometric coherence due to the strong target samples being excluded.

In Fig. 16, we show the bathymetry estimate using 9×9 square window averaging in Fig. 16(a), 3×3 square window averaging in Fig. 16(b), and our image segmentation filtering in Fig. 16(c). The red circles outline the assumed target positions as in the other figures.



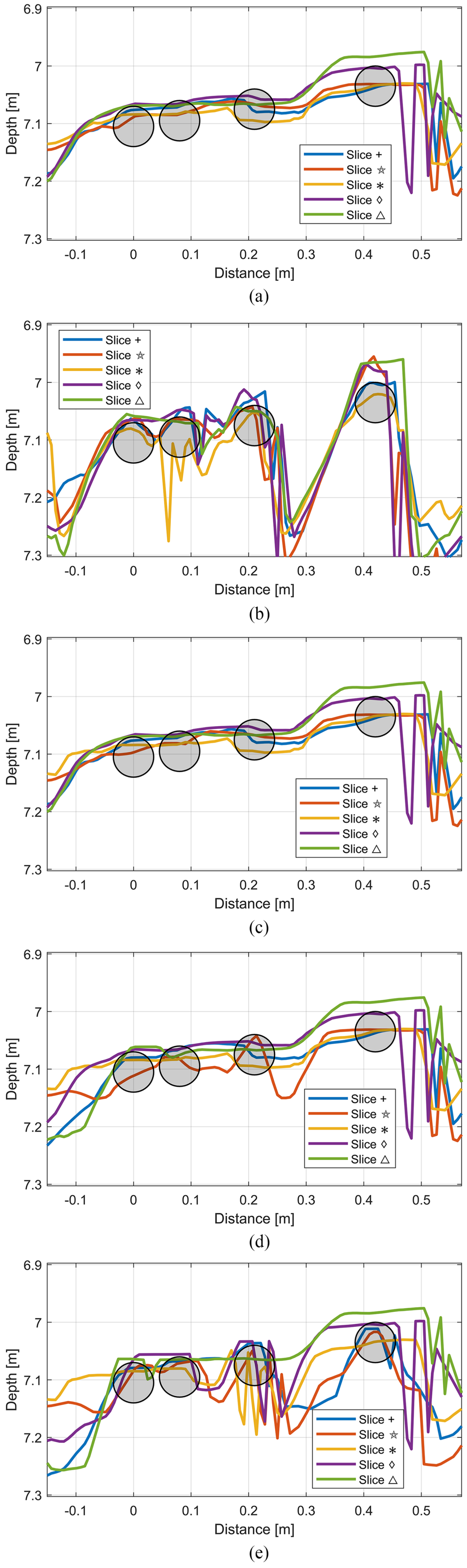
(a)

(b)

(c)

Fig. 16. Estimated bathymetries projected in ground plane. The black circles are target outlines manually superimposed on top of the images as a guide, but do not represent ground truth positions of the targets. The black lines indicate the slice positions used in Figs. 17 and 18. (a) Square 9 × 9. (b) Square 3 × 3. (c) Segmentation, four segments.

In Figs. 17 and 18, we show plots of the solid and dashed slices, respectively, as illustrated in Fig. 16. The small spacings across the slices in Fig. 17 is not resolved with the 9×9 window, but can be partially resolved by the 3×3 window at the cost of higher variance. We observe that our method provides better resolution than the 9×9 window. The improvement is more evident with the slightly larger spacings in Fig. 18, where we clearly observe shapes better representing the ground truth sphere sizes, while maintaining much lower variance than with



(a)

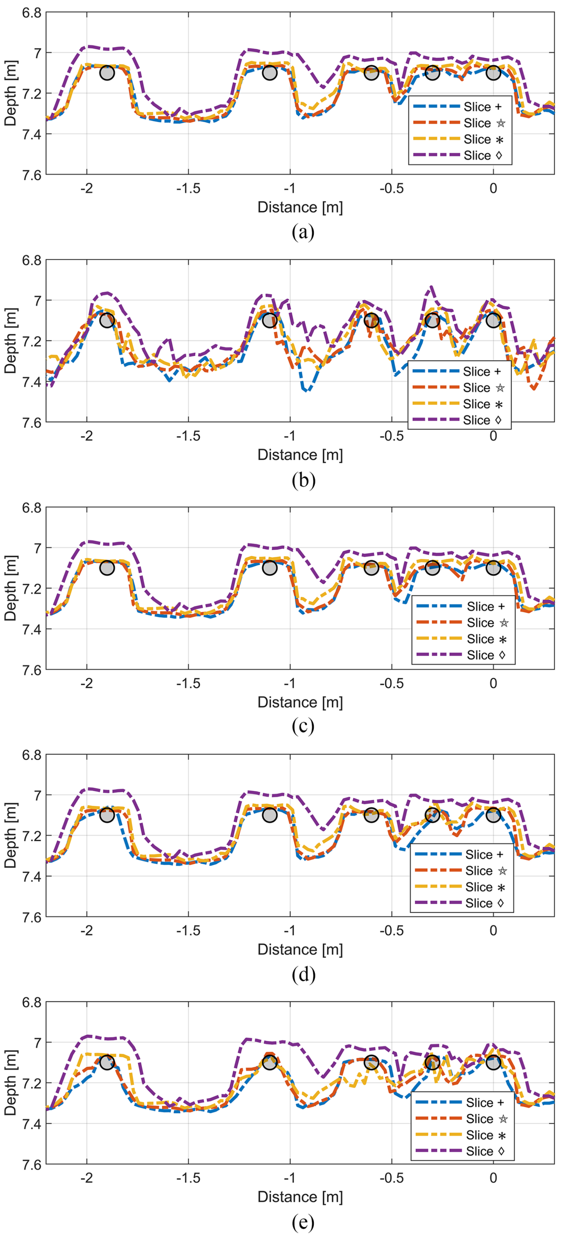
(b)

(c)

(d)

(e)

Fig. 17. Bathymetry slices (solid lines) across the objects for various processing options. The superimposed circles indicates the manually estimated target positions, constrained by the known target size and spacing. (a) Square 9×9. (b) Square 3×3. (c) Segmentation, two segments. (d) Segmentation, three segments. (e) Segmentation, four segments.



(a)

(b)

(c)

(d)

(e)

Fig. 18. Bathymetry slices (dashed lines) across the objects for various processing options. The superimposed circles indicates the manually estimated target positions, constrained by the known target size and spacing. (a) Square 9 × 9. (b) Square 3 × 3. (c) Segmentation, two segments. (d) Segmentation, three segments. (e) Segmentation, four segments.

the 3 × 3 window. The difference in variance might seem slight between our method and the 3 × 3 window when inspecting these slices, but comparing the larger bathymetry plot in Fig. 16 we see that the 3 × 3 window maintains this high variance also in areas with flatter seafloor, while our method performs equally to the 9 × 9 window method in these areas where no segments were detected from the intensity image.

H. Real Data Discussion

We find that the segmentation results in Fig. 14 are acceptable based on the image data, but only some of the spheres are resolved. This could potentially have been improved by using a more advanced segmentation algorithm, but after inspecting the SAS image, we consider this to be nontrivial. To avoid introducing any new parameters at this point, we therefore kept the same, automated algorithm as we used in the simulation case. Any improvements to the segmentation algorithm would improve our method, but further studies on the image segmentation task is outside of the scope of this study.

In Fig. 15, we found a reduction of the number of samples near segment edges, and consequently we observed a reduction in coherence around the targets. As we saw with the simulated case in Section III-B, this may be a less biased estimate of the depth in these positions, since we were no longer perturbed by the strong signal from the nearby targets. Similarly, we observed that the Cramer-Rao lower bound (CRLB) for the depth estimate, which expresses the lower bound on the variance of an unbiased estimator, remained equally low on the targets even with fewer samples because of the higher SNR. As for the surrounding seafloor, we observed a higher CRLB due to the exclusion of the strong target samples.

In Fig. 16(a), the bathymetry estimates are unable to resolve any of the spheres properly, as is expected since the horizontal resolution is worse than the sphere separation. With bathymetry estimated by a smaller window in Fig. 16(b), we were able to resolve the spheres better, but at the cost of increased variance in the bathymetry estimate. Using our suggested method to estimate the bathymetry in Fig. 16(e), we were able to maintain low variance comparable to the results using 9x9 square windows, while still separating the spheres with largest distance and partly the second-largest distance as well. We observe that the seafloor surrounding the targets is estimated closer to what we would expect considering our knowledge about the target and the surrounding seafloor.

In Figs. 17 and 18, we plotted the slices indicated in Fig. 16 with the manually estimated sphere positions as a reference. We were not able to resolve all the spheres, but inspecting the segmentation plot in Fig. 14 this is to be expected based on the segmentation result. However, we were able to resolve the spheres significantly better than using the 9 × 9 window, both with respect to resolution and the shape resolvability. Our result was closer to estimating the correct extent of the spheres. Again, we find that the segmentation algorithm is the most important part of our method. However, regardless of imperfect segmentation results we find that the resulting bathymetry using our method improves on using square windows, either by improved accuracy and resolution, or by lower variance.

IV. Conclusion

We introduced a novel method using SAS image segmentation maps to constrain the averaging windows when doing sample averaging in SAS interferometric processing. We showed that we can achieve more accurate object rendering and better resolution for object separation with our method, while maintaining the low variance of a sufficiently large averaging window. With the simulated data set, we demonstrated a 30% improvement in RMSE for the along-track case. We achieved similar results with real data, demonstrating an improved ability to better resolve small objects.

The proposed method relies heavily on the segmentation map, and thus the segmentation algorithm is a critical part of the processing chain. We used a simple algorithm which serves well to demonstrate our method. However, replacing this algorithm with more advanced methods may significantly improve the performance of the proposed method.

The main disadvantages of our method were related to the segmentation. For imprecise segmentation maps, the resulting bathymetry may contain artifacts related to the segment shapes. However, as we have shown with small, protruding objects, these errors will often be smaller than the errors induced by large averaging windows. We therefore claim that our method of constraining the averaging window is an improvement compared to using the larger fixed-size windows.

Using our suggested method, the number of image samples used to form each bathymetric estimate varied, and thus the variance of the bathymetric estimates were affected. For future work, we would consider extending the method to adapt the window shape to make the window size approach the same size as the full window whenever possible. This is possible as long as the segment is sufficiently large.

We showed that using the intensity image for segmentation is a sound approach because of its relation to the geometry of the seafloor. On both simulated and real data, we showed improved conservation of object shapes as well as improved horizontal resolution of closely spaced targets with our suggested method.

Acknowledgment

The authors would like to acknowledge the Bundeswehr Technical Center for Ships and Naval Weapons, Maritime Technology and Research (WTD 71), and Hauke Voss in particular, for the sonar experiment and data set used in this study. Furthermore, we would like to thank Andreas Austeng, University of Oslo, and Øyvind Ødegård, the Norwegian University of Science and Technology, as well as Stig A. V. Synnes and Marc Geilhufe at the Norwegian Defence Research Establishment (FFI) for their helpful input and discussions related to this study.

References

[1] H. D. Griffiths, T. A. Rafik, Z. Meng, C. F. N. Cowan, H. Shafeeu, and D. K. Anthony, "Interferometric synthetic aperture sonar for high-resolution 3-D mapping of the seabed," *IEE Proc. Radar Sonar Navigation*, vol. 144, no. 2, pp. 96–103, Apr. 1997.

[2] R. F. Hanssen, *Radar Interferometry: Data Interpretation and Error Analysis (Remote Sensing and Digital Image Processing)*. Berlin, Germany: Springer, 2001.

[3] T. O. Sæbø, "Seafloor depth estimation by means of interferometric synthetic aperture sonar," Ph.D. dissertation, Faculty Sci. Technol. Dept. Phys. Technol., Univ. Tromsø, Tromsø, Norway, 2010.

[4] M. P. Hayes and P. T. Gough, "Synthetic aperture sonar: A review of current status," *IEEE J. Ocean. Eng.*, vol. 34, no. 3, pp. 207–224, Jul. 2009.

[5] R. Bamler and P. Hartl, "Synthetic aperture radar interferometry," *Inverse Probl.*, vol. 14, no. 4, Aug. 1998, Art. no. R1.

[6] A. Denny, R. E. Hansen, T. O. Sæbø, and R. B. Pedersen, "The use of synthetic aperture sonar to survey seafloor massive sulfide deposits," *J. Ocean Technol.*, vol. 10, pp. 36–53, 2015.

[7] Ø. Ødegård, R. E. Hansen, H. Singh, and T. J. Maarleveld, "Archaeological use of synthetic aperture sonar on deepwater wreck sites in skagerrak," *J. Archaeol. Sci.*, vol. 89, pp. 1–13, Jan. 2018.

[8] J. T. Christoff, J. E. Fernandez, and D. A. Cook, "Unmanned underwater vehicle broadband synthetic aperture sonar," in *Proc. OCEANS '02 MTS/IEEE*, 2002, pp. 1871–1877.

[9] D. Williams, "SAS and bathymetric data fusion for improved target classification," in *Proc. Int. Conf. Underwater Remote Sens*, 2012, pp. 123–135.

[10] V. Myers and J. Fawcett, "A template matching procedure for automatic target recognition in synthetic aperture sonar imagery," *IEEE Signal Process. Lett.*, vol. 17, no. 7, pp. 683–686, Jul. 2010.

[11] J. Del Rio Vera, E. Coiras, J. Groen, and B. Evans, "Automatic target recognition in synthetic aperture sonar images based on geometrical feature extraction," *EURASIP J. Adv. Signal Process.*, vol. 2009, Apr. 2009, Art no. 109438.

[12] O. J. Lorentzen, T. O. Sæbø, and R. E. Hansen, "Backscatter features for estimating synthetic aperture sonar bathymetry," in *Global Oceans 2020.* Singapore–U.S. Gulf Coast, Oct. 2020.

[13] G. Xu, Y. Gao, J. Li, and M. Xing, "InSAR phase denoising: A review of current technologies and future directions," *IEEE Geosci. Remote Sens. Mag.*, vol. 8, no. 2, pp. 64–82, Jun. 2020.

[14] J.-S. Lee, K. P. Papathanassiou, T. L. Ainsworth, M. R. Grunes, and A. Reigber, "A new technique for noise filtering of SAR interferometric phase images," *IEEE Trans. Geosci. Remote Sens.*, vol. 36, no. 5, pp. 1456–1465, Sep. 1998.

[15] M. S. Seymour and I. G. Cumming, "Maximum likelihood estimation for SAR interferometry," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 1994, pp. 2272–2275.

[16] R. M. Goldstein and C. L. Werner, "Radar interferogram filtering for geophysical applications," *Geophys. Res. Lett.*, vol. 25, no. 21, pp. 4035–4038, Nov. 1998.

[17] I. Baran, M. P. Stewart, B. M. Kampes, Z. Perski, and P. Lilly, "A modification to the goldstein radar interferogram filter," *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 9, pp. 2114–2118, Sep. 2003.

[18] R. Song et al., "Improved goldstein SAR interferogram filter based on adaptive-neighborhood technique," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 1, pp. 140–144, Jan. 2015.

[19] Z. Suo, J. Zhang, M. Li, Q. Zhang, and C. Fang, "Improved InSAR phase noise filter in frequency domain," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 2, pp. 1185–1195, Feb. 2016.

[20] A. Mestre-Quereda, J. M. Lopez-Sanchez, J. Selva, and P. J. Gonzalez, "An improved phase filter for differential SAR interferometry based on an iterative method," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 8, pp. 4477–4491, Aug. 2018.

[21] G. Vasile, E. Trouve, J.-S. Lee, and V. Buzuloiu, "Intensity-driven adaptive-neighborhood technique for polarimetric and interferometric SAR parameters estimation," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 6, pp. 1609–1621, Jun. 2006.

[22] R. E. Hansen, "Introduction to synthetic aperture sonar," in *Sonar Systems*. London, U.K.: IntechOpen, 2011.

[23] D. H. Johnson and D. E. Dudgeon, Array signal processing: Concepts and techniques. Saddle River, NJ, USA: Prentice Hall, 1993.

[24] C. Oliver and S. Quegan, *Understanding Synthetic Aperture Radar Images, Ser. Artech House Remote Sensing Library*. Norwood, MA, USA: Artech House, 1998.

[25] D. Arthur and S. Vassilvitskii, "K-means : The advantages of careful seeding," in *Proc. Annu. ACM-SIAM Symp. Discrete Algorithms*, 2007, pp. 1027–1035.

[26] A. Buades, B. Coll, and J.-M. Morel, "Non-local means denoising," *Image Process. On Line*, vol. 1, pp. 208–212, Sep. 2011.

Ole Jacob Lorentzen (Member, IEEE) received the M.Sc. degree in informatics, in 2012, from the University of Oslo, Oslo, Norway, where he is currently working toward the Ph.D. degree in synthetic aperture sonar interferometry in rough seafloor bathymetry with the Department of Informatics.

Since 2013, he has been with the Norwegian Defence Research Establishment (FFI), Kjeller, Norway, working in the fields of sonar signal processing, synthetic aperture sonar and autonomous underwater vehicle operations. He is currently a Senior Scientist with FFI.

Torstein Olsmo Sæbø (Senior Member, IEEE) was born in Bergen, Norway, in 1977. He received the M.Sc. degree (or cand.scient) in astrophysics from the University of Oslo, Oslo, Norway, in 2002, and the Ph.D. degree in physics from the University of Tromsø, Tromsø, Norway, in 2010, with a dissertation entitled "Seafloor depth estimation by means of interferometric synthetic aperture sonar." Since 2002, he has been with the Norwegian Defence Research Establishment (FFI), Kjeller, Norway, specializing in the field of interferometry on synthetic aperture sonar. He is currently a Principal Scientist with FFI.

Dr. Sæbø was Research Manager for the Underwater Robotics Research Group at FFI. He was an Associate Editor for *IEEE Journal of Oceanic Engineering*.

Alan J. Hunter (Senior Member, IEEE) received the B.E. (Hons.) and Ph.D. degrees in electrical and electronic engineering from the University of Canterbury, Christchurch, New Zealand, in 2001 and 2006, respectively.

From 2007 to 2010, he was a Research Associate with the University of Bristol, Bristol, U.K., and from 2010 to 2014, he was a Defense Scientist with TNO (Netherlands Organisation for Applied Scientific Research), Hague, The Netherlands. In 2014, he joined the Faculty of Engineering, University of Bath, Bath, U.K., where he is currently an Associate Professor and Deputy Head of Department. Since 2017, he has been an Adjunct Associate Professor with the Department of Informatics, University of Oslo, Oslo, Norway. His research interests include underwater acoustics, sonar imaging of the seafloor, and autonomous underwater systems.

Dr. Hunter is an Associate Editor for *IEEE Journal of Oceanic Engineering*.

Roy Edgar Hansen (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in physics from the University of Tromsø, Norway, in 1992 and 1999, respectively.

From 1992 to 2000, he was with the Norwegian research company TRIAD, working on multistatic sonar, multistatic radar, SAR, and underwater communications. Since 2000, he has been with the Norwegian Defence Research Establishment (FFI), Kjeller, Norway, working in the field of synthetic aperture sonar. He is currently Principal Scientist with FFI. He is also an Adjunct Professor in acoustic imaging with the Department of Informatics, University of Oslo, Oslo, Norway. His research interests include synthetic aperture sonar and radar, ultrasound imaging, sonar signal processing, and array signal processing.