Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Saliency and Anomaly: Transition of Concepts from Natural Images to Side-Scan Sonar Images ⋆

Nadir Kapetanović * **Nikola Mišković** * **Adnan Tahirović** **

* *University of Zagreb, Faculty of Electrical Engineering and Computing, Zagreb, Croatia (e-mail: nadir.kapetanovic@fer.hr, nikola.miskovic@fer.hr).*
** *Faculty of Electrical Engineering, University of Sarajevo, Sarajevo, Bosnia and Herzegovina (e-mail: adnan.tahirovic@etf.unsa.ba)*

Abstract: An autonomous underwater vehicle (AUV) or a multi-AUV system performing autonomous seafloor exploration missions by a side-scan sonar need to perceive their environment in order to replan the mission if they detect interesting objects in sensor data. Several anomalous/salient object detection methods mostly used for natural images are here applied to sonar images. All methods were firstly benchmarked on a 1500 simulated side-scan sonar images dataset. Precision-recall and processing time analysis was conducted in order to choose the best suited method in such controlled conditions. The performance of the best performing detection method was then validated on a 350 real side-scan sonar images dataset. This method was then implemented and optimized for the computer onboard an AUV. It turned out to be fast enough for online processing of large volumes of sonar data.

Keywords: side-scan sonar, image processing, anomaly detection, target detection, saliency

1. INTRODUCTION

Side-scan sonar is often used in marine monitoring and exploration missions. This includes exploration of biosphere, exploration of underwater archaeological sites, marine safety, and many other applications. Side-scan sonar survey missions are executed either by a tethered towfish equipped with a side-scan sonar, Plets et al. (2013), or by using remotely operated vehicles (ROVs) or autonomous underwater vehicles (AUVs), Ludvigsen et al. (2014). Deploying a tethered towfish from a boat requires hiring a boat and its crew: a towfish operator, a side-scan operator, and of course experts who can interpret sonar data online and tag interesting objects to be revisited later on in a high detail mission.

Deploying an AUV is another option. Since communication bandwith is a significant constraint on the AUV operations and fast transmission of large sonar data volumes is still not possible, sonar data is typically stored on board the vehicle. This introduces an inability of on-the-fly manual high detail mission (re)planning by the sonar data interpreter, since all the sonar data can be processed only after the initial survey mission, Chapple (2008).

The best option is to deploy an AUV with side-scan sonar which would autonomously cover some area of seafloor, i.e. process sonar data online and autonomously make decisions how to replan the mission to record in more detail some interesting objects detected in sonar imagery. Automatic detection and classification techniques are being developed for several reasons: to provide reliable and consistent detection of objects on the seabed; to free human analysts from time-consuming and tedious detection tasks; and to enable autonomous in-field decision-making and mission (re)planning based on observations of the detected objects, Chapple (2008).

In authors' previous work, several online side-scan sonar data-driven coverage path planning (CPP) algorithms were proposed for monitoring and surveying large-scale (over $1km^2$) seafloor regions by an AUV and published in Kapetanović et al. (2019). One of the task execution algorithms needed for AUV to autonomously map some area of interest is a module which detects interesting objects in side-scan sonar data. For a given seafloor area's side-scan sonar imagery, this module should output the coordinates of the interesting objects (if detected) in the area and send them to the coverage path planning module which then replans the AUV's path in order to inspect those objects in more detail. The goal is to implement all the these modules for Lupis AUV, shown in Fig. 1, which Laboratory for Underwater System and Technologies (LABUST) acquired by from OceanScan - MST company.

Considering side-scan pings stacked together (in a so-called "waterfall view") as a grayscale image, in this paper the authors implemented and tested various methods for interesting object detection. Contrast-based saliency

⋆ This research is sponsored by Croatian Science Foundation Multi Year Project under G.A. No. IP-2016-06-2082 named CroMarX; the Foundation of the Croatian Academy of Science and Arts and the EU Regional Development funded project DATACROSS under G.A. No. KK.01.1.1.01.0009; and the H2020-INFRAIA funded EUMarineRobots project under G.A. No. 731103.

Fig. 1. AUV Lupis

method in Zhai and Shah (2006), Itti-Koch saliency in Itti et al. (1998), and SIMP-SAL saliency in Harel (2019 (accessed February, 2019) as a simplification of Itti-Koch method, are almost exclusively used for natural images. The anomaly detection method in Kaeli (2016), and graph-based visual saliency (GBVS) saliency in (Harel et al., 2006) were, however, used on side-scan sonar data. In this paper, contrast based saliency method from Zhai and Shah (2006), originally used on single-scale natural images, is extended for use on multiple resolution scales of side-scan sonar images. Also, processing time of the anomaly detection method from Kaeli (2016) is significantly improved.

The above mentioned methods were benchmarked in Matlab on a dataset of 1500 simulated side-scan sonar images containing a single and also multiple objects of various sizes and at various positions. The anomaly detection method from Kaeli (2016) had the best precision and recall performance on simulated sonar images dataset. Its performance was then validated on a dataset of 350 real side-scan sonar images and it also had satisfactory recall-precision performance. After prototyping and performance testing phase, the method was implemented using OpenCV library and integrated into Robot Operating System (ROS) environment in order to benchmark its processing time on the computer onboard AUV Lupis. It turned out to be fast enough to process 1 megapixel (MP) in $1.5 - 2s$ which is fast enough for large volumes of sonar data being recorded.

The rest of the paper is organized as follows: a short introduction to the terms saliency and anomaly is given in Section 2. Stateflow developed for using saliency detection methods on side-scan sonar images is given in Section 3. Section 4 presents the performance metrics used to assess how well do the mentioned methods detect salient/anomalous objects in simulated side-scan sonar images. The best performing method from Section 4 is then validated on real side-scan sonar image dataset in Section 5. This method is reimplemented to run on the target hardware of AUV Lupis, and its processing time, crucial for online applications, is analyzed in Section 6. Concluding remarks are given in Section 7.

2. OVERVIEW OF SALIENCY METHODS

In order to design algorithms for interesting objects detection in sonar images, it is first needed to understand how human visual system notices interesting things which pop up from the rest of the visual field. Our visual system is selective, i.e., we concentrate on certain aspects of a scene while neglecting other things. It is interesting to note that human brain uses a tiny fraction (<1%) of the collected visual information from the optic nerve to build a representation of the environment. In the literature, two main attention mechanisms are discussed: top-down and bottom-up. Top-down is voluntary, goal-driven and slow, i.e., typically in the range between $100ms$ and several seconds. In contrast, bottom-up attention (also known as *visual saliency*) is associated with attributes of a scene that draw the attention to a particular location. These attributes include motion, contrast, orientation, brightness and color. Bottom-up mechanisms are involuntary, and faster than top-down, Sharma (2015). In the past few decades, modeling of visual saliency has generated a lot of interest in the research community. It has paved the way for a number of computer vision applications such as: target detection, image and video compression, image segmentation, context aware image resizing, robot localization, image retrieval, image and video quality assessment, dynamic lighting, advertisement, artistic image rendering and human–robot interaction Sharma (2015).

On the other hand, similarly, *anomaly detection* refers to the problem of finding patterns in data that do not conform to expected behavior. The importance of anomaly detection is due to the fact that anomalies in data translate to significant (and often critical) actionable information in a wide variety of application domains, Chandola et al. (2009). It is thus safe to say that interesting objects in sonar imagery are both salient and anomalous w.r.t. the surrounding mostly sandy seafloor.

Two broad classes of salient/anomalous object detection/classification algorithm are in use: *supervised algorithms*, requiring training data with target objects in known locations, and *unsupervised algorithms*. Unsupervised saliency map and anomaly detection algorithms applied in the field of computer vision use a number of models, i.e. Bayesian, cognitive, decision theoretic, graphical, information theoretic, pattern classification, spectral analysis, and many other types of models. Interested reader is referred to (Sharma (2015)) and (Chandola et al. (2009)) survey papers on unsupervised saliency and anomaly detection methods. Supervised saliency map and anomaly detection algorithms are implemented as artificial neural networks (ANNs) which need exhaustive training data sets to have good performance and generalization under arbitrary circumstances as in Ji et al. (2018) and Huang et al. (2015).

Techniques for computer-aided detection/ classification (CAD/ CAC) in sidescan sonar imagery are under development since the early 1990s. The most successful techniques rely on the presence of a coupled acoustic highlight and shadow associated with an object sitting on the seabed, Chapple (2008). In the context of detecting interesting objects or targets in side-scan sonar images, several saliency/anomaly detection methods have been developed, both unsupervised as well as supervised, e.g. Reed et al. (2003), Mishne et al. (2015), Zhu et al. (2019), Goldman and Cohen (2004), Noiboar and Cohen (2007), Zhu et al. (2017), Kaeli (2016), Mishne and Cohen (2013).

3. STATEFLOW OF THE INTERESTING OBJECT DETECTION METHODS ADAPTED FOR SIDE-SCAN SONAR IMAGES

In this paper five anomalous/salient object detection methods were implemented and tested. Contrast-based saliency method (CON-SAL) from Zhai and Shah (2006),

graph-based visual saliency (GBVS-SAL) from (Harel et al., 2006), Itti-Koch saliency (ITTI-SAL) from Itti et al. (1998), and SIMP-SAL saliency from Harel (2019 (accessed February, 2019) as a simplification of Itti-Koch method are usually used for natural images. However, the anomaly detection method (ANOMALY) from Kaeli (2016) was used exclusively for side-scan sonar imagery. It is worth noting that other methods have also been tried out, namely Mishne and Cohen (2013), which uses diffusion maps for anomaly detection, but was too slow to be considered for online use on-board an AUV, even though it detected anomalous interesting object quite good. Method for generating image signatures based on highlighting sparse salient objects in an image, Hou et al. (2012), was much worse than the other methods when used on side-scan sonar images. Also, translating side-scan sonar grayscale image (or its parts) into its fractal dimension to differ naturally occurring objects and man-made objects based on method Liu et al. (2014) was a few orders of magnitude slower than the other methods even for small portions of sonar images.

The reason why artificial neural networks (ANNs) were not considered as a potential method for anomaly detection was that problem of anomalous object detection is much more general than classification problem, and the dataset of 350 real side-scan sonar transect images that was available to the authors was in their opinion insufficient training set which may lead to the lack of generalization capabilities of the ANN. Also, simulated side-scan sonar images could most probably not cover many variations of seafloor types and objects, which would again lead to an insufficiently diverse training dataset.

A generic stateflow was developed in order to use any of the five methods on any side-scan sonar image. Pseudocode of this stateflow is given in Algorithm 1. All of the above mentioned methods process side-scan sonar image *img* which is pre-proceesed by removing nadir part (line 2 in Algorithm 1). Then an along- and across-track normalization is performed by a moving average filter (of length *params.mavg_M*) to compensate for the attenuation of the returned signal towards the edges of the sonar range (line 3 in Algorithm 1). After that, the image is smoothed by Gaussian filter of kernel *params.gauss_kernel* to additionally suppress noise (line 4 in Algorithm 1). The saliency/anomaly detection method *params.sal_method* (any of: ITTI-SAL, CON-SAL, ANOMALY, GBVS-SAL, or SIMP-SAL) is then applied to the pre-processed sonar image (line 5 in Algorithm 1).

After the saliency/anomaly map $salMap$ is computed, it is then thresholded w.r.t. its mean value *params.saliency_th* (line 6 in Algorithm 1). Edge detection is then applied to extract all edges of the most salient areas in the sonar image (line 7 in Algorithm 1). Those edges are then thinned down to remove a few pixel objects by a median filter defined by *params.median_kernel* (line 8 in Algorithm 1). The edges are then segmented into separate contours (line 9 in Algorithm 1). Contours which enclose area less than a predefined parameter $params.area_{th}$ are then discarded (line 10 in Algorithm 1). Next step discards the contours whose interior pixels' mean brightness is less than a predefined parameter $params.brig_{th}$ (line 11 in Algorithm 1). Finally, the oriented bounding boxes of the remaining

Algorithm 1 Stateflow for anomalous object detection

1: **function** DETECTANOMALIES(*img*, *params*)
2: $img \leftarrow img$.**removeNadir**
3: $img \leftarrow img$.**normalizePings**($params.mavg\_M$)
4: $img \leftarrow img$.**smooth**($params.gauss\_kernel$)
5: $salMap \leftarrow img$.**getSalMap**($params.sal\_method$)
6: $salMap \leftarrow salMap$.**threshold**($params.sal\_th$)
7: $edges \leftarrow salMap$.**detectEdges**()
8: $edges \leftarrow edges$.**median**($params.median\_kernel$)
9: $contours \leftarrow edges$.**segment**()
10: $contours \leftarrow contours$.**threshold**($params.area_{th}$)
11: $contours \leftarrow contours$.**threshold**($params.brig_{th}$)
12: $bboxes \leftarrow contours$.**getBBoxes**($contours$)
13: **return** *bboxes*

contours are computed for easier representation (line 14 in Algorithm 1). The results this stateflow's performance testing are given in the following sections.

Pseudocode of the ANOMALY method from Kaeli (2016), here named *getAnomalyMap* method, is given in Algorithm 2, and is called in Algorithm 1 in line 5 as *img.getSalMap('ANOMALY')*. This algorithm scales down a sonar image to $params.n_{scales}$ resolution levels (line 4 in Algorithm 2) and each time applies Laplacian of Gaussian (LoG) filtering (line 5 in Algorithm 2). Then it scales up the image to its original size (line 6 in Algorithm 2) and normalizes it w.r.t. to its mean value (line 7 in Algorithm 2). The resulting image is stored into a so-called filter bank *f_bank* (line 8 in Algorithm 2). After the filter bank has been computed, the anomaly map $salMap$ of local pixel histogram differences is computed. The histogram difference metric for local variation of a pixel w.r.t. its neighbors (lines $9-11$ in Algorithm 2), defined in Kaeli (2016) as ℓ_1 norm, is that algorithm's serious bottleneck. It is here approximated as an Nd kernel $diff_{kernel}$ (lines $9-10$ in Algorithm 2). The resulting $salMap$ is computed as a convolution of the filter bank *f_bank* with the kernel $diff_{kernel}$, which gives similar results to the original ℓ_1 norm in Kaeli (2016), but in a fraction of the time.

CON-SAL method is here extended for use on side-scan sonar images and also on multiple resolution levels. It is called in Algorithm 1 in line 5 as *img.getSalMap('CON-SAL')*. Pseudocode of the CON-SAL method is almost the same as Algorithm 2. The only difference between CON-SAL and ANOMALY method is in line 5 of Algorithm 2. Instead of LoG filtering, the extended CON-SAL method computes the brightness distance map.

Algorithm 2 Multiscale anomaly map

1: **function** GETANOMALYMAP(*img*,*params*)
2: $f\_bank \leftarrow zeros(img.size(), params.n_{scales})$
3: **for** $i \leftarrow 1$ to $params.n_{scales}$ **do**
4: $img_{sd} \leftarrow img$.**scaleDown**($params.scales[i]$)
5: $img_{log} \leftarrow img_{sd}$.**LoG**($params.log\_kernel$)
6: $img_{su} \leftarrow img_{sd}$.**scaleUp**($params.scales[i]$)
7: $img_{su} \leftarrow img_{su}$.**normalize**()
8: $f\_bank[i] \leftarrow img_{su}$
9: $diff_{kernel} = ones(3, 3, params.n_{scales})$
10: $diff_{kernel}(2, 2, :) = -8$
11: $salMap \leftarrow$ **convolve**($f\_bank, diff_{kernel}$)
12: **return** $salMap$

4. SIMULATED DATASET RESULTS

The first step to deciding which method is best used for the problem at hand, a simple side-scan sonar images dataset of 1500 $2MP$ images was generated, mostly as noisy grayscale images with a few $(1-5)$ objects and their simulated sonic shadows of various sizes $(1x1, 1.5x1.5, ..., 10x10m^2)$ and position distributions in the sonar image. With the assumed $50m$ across-track range of the side-scan sonar, the resolution of each pixel was taken as $5 \times 5cm$. Performance benchmarking of the above mentioned five methods for salient/anomalous object detection was performed on a computer with a quad-core $2.8GHz$ CPU with $16GB$ of RAM and $4GB$ GPU. Software used for benchmarking on the simulated dataset was MATLAB with its Image Processing Toolbox and Parallel Computing Toolbox with CUDA GPU Processing Support. All five methods used to detect interesting objects in side-scan sonar images were optimized for execution on a GPU in order to perform as fast as possible. This was done having in mind that these methods would also be used in Gazebo simulator in which AUV simulator, Manhães et al. (2016), and side-scan sonar simulator, Bore et al. (2019), would be integrated with the sonar image processing module, the online coverage path planning module, Kapetanović et al. (2019), as well as the control module, Kapetanović et al. (2017).

All five methods were applied to the simulated side-scan sonar image dataset, and were compared to the known ground truth of interesting objects pixels indices. An example of such simulated side-scan sonar image and five methods' performance in detecting interesting objects standing out in those images is given in Fig. 2. It is clear that the anomaly detection method from Kaeli (2016) gives the most accurate and precise detection results compared to other methods. In order to formalize this, a precision-recall analysis was conducted. The so-called F harmonic mean metric (Sokolova and Lapalme (2009)) was used to aggregate these two metrics, which is defined as:

$$F = 2 \times \frac{precision \times recall}{precision + recall}. \quad (1)$$

Results from all 1500 tests conducted are shown in Fig. 3. Mean values of recall-precision performance of all 5 methods are given in Fig. 3a. It is clearly visible that at 93% F-measure, the anomaly detection method from Kaeli (2016) is much better than Itti-Koch, GBVS, as well as contrast based saliency method. SIMP-SAL is the closest with 76% F-measure. Fig. 3b shows execution times of all 1500 runs for each of the chosen 5 methods normalized by the image size in MP. It is obvious that SIMP-SAL is much faster than all other benchmarked methods, and anomaly detection method is second fastest, but significantly more precise and accurate in its detections. Taking into account that the target hardware on-board AUV Lupis is roughly $10-50$ times slower than a high performance workstation computer used for benchmarking, even then anomaly detection method running $2-10s/MP$ is considered by the authors to be fast enough taking into account its winning precision/recall performance as well.

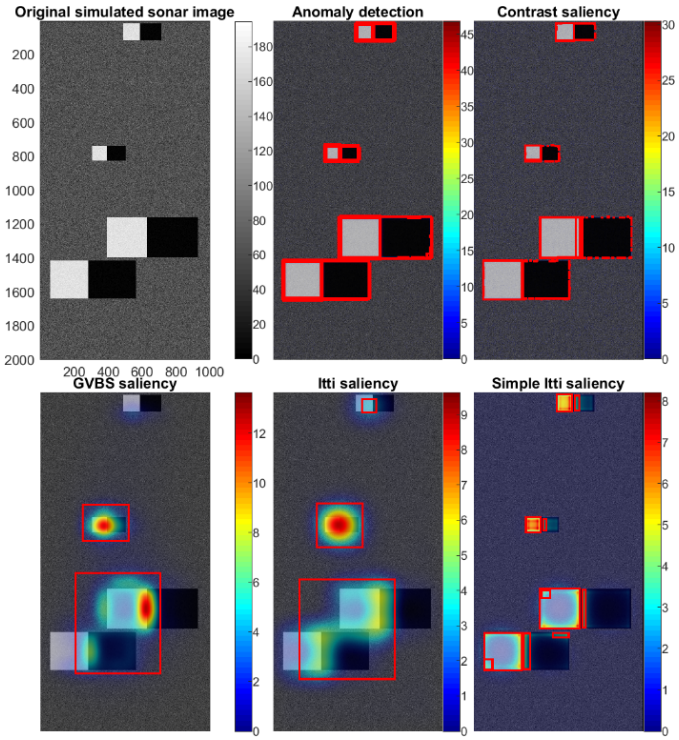
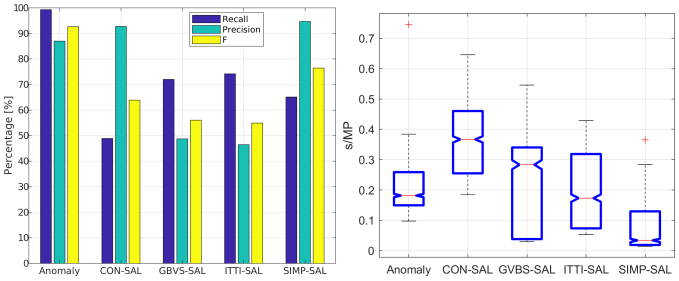


Fig. 2. An example of interesting objects detection in simulated side-scan sonar data by interesting object detection methods. Colorbar represents each saliency metric normalized by its mean value to visualize which areas are the most interesting. Since anomaly and contrast methods practically detect edges in multiple scales, the anomaly/saliency values plots in the upper subfigures are covered with detection bounding boxes.



(a) (b)

Fig. 3. (a) Comparison of mean recall, precision, and combined metric usually used in interesting object detection methods. (b) Distributions of processing times normalized by sonar image size.

5. REAL DATASET RESULTS

To the best of authors' knowledge, standardized benchmarks for object detection specifically in the case of side-scan sonar imagery do not (yet) exist. Thus, in order to validate the results mentioned in Section 4, numerous real side-scan sonar datasets were acquired from field trials with the AUV in mostly underwater archealogical sites near Cavtat, Croatia, Baiae Bay, Italy, Peristera island, Kikinthos island, Glaros and Tilegraphos Capes in Greece, as well as around Pelješac Penninsula and Biograd na Moru in Croatia. Anomaly detection method from Kaeli (2016) as the best method on simulated data was applied

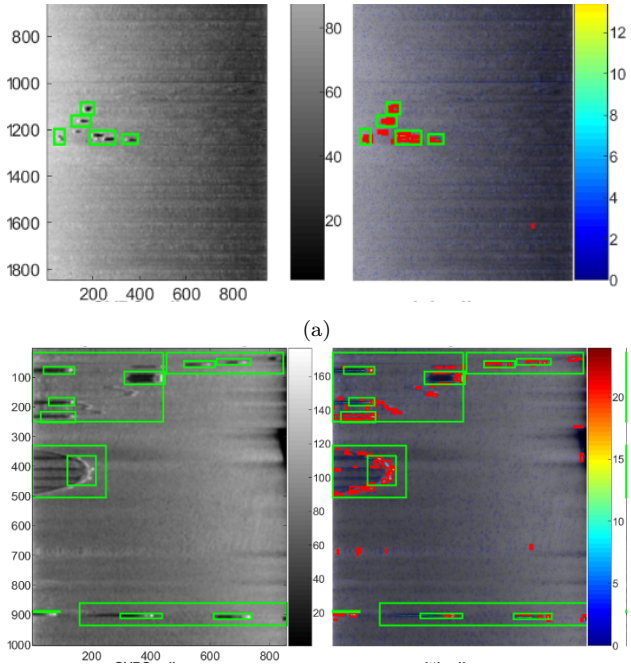


Fig. 4. An example of interesting objects detection in real side-scan sonar data by interesting object detection methods. Colorbar represents each saliency metric normalized by its mean value to visualize which areas are the most interesting. Since anomaly detection method Kaeli (2016) detects edges in multiple scales, the anomaly values plot is covered with detection bounding boxes.

to this dataset and using the same computer as for the simulated dataset.

The resulting interesting objects detection results are given in Fig. 4. The ground truth of what is interesting in the given sonar images was obtained from human operators circling the objects standing out from the ususal clutter and noise in the side-scan sonar images. It can be noted that the anomaly detection method gives intuitive results for interesting objects' detection in the real side-scan sonar data as well, i.e. of all five methods it matches the human perception of "salient" and "outstanding" objects in the noisy side-scan sonar data the best.

6. RUNNING ON TARGET HARDWARE

After prototyping and performance testing phase, the anomaly detection method was implemented using OpenCV library and integrated into Robot Operating System (ROS) environment in order to benchmark its processing time on the target hardware onboard AUV Lupis, namely UDOO DUO. UDOO has a dual-core $1GHz$ CPU, $1GB$ of RAM, and a Vivante GPUs for 2D, 3D and vector graphics.

Additional changes were made to the OpenCV implemented anomaly detection algorithm, namely converting the Nd convolution described in Section 3 into N $2D$ convolutions for computing local pixel histogram difference through the whole filter bank, since they are linearly separable and OpenCV did not offer a straightforward Nd convolution method. Also, Contrast-Limited Adaptive

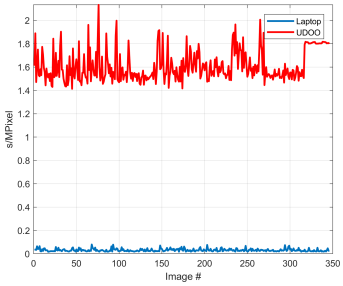


Fig. 5. Unit processing time per megapixel UDOO vs. computer.

Histogram Equalization (CLAHE) algorithm, Pizer et al. (1990), was used to normalize the brightness of the sonar images, thus improving the contrast at image edges.

Processing time for anomalous/salient object detection algorithm on UDOO is on average $50-60$ times slower than the processing time for anomaly detection on the computer, see Fig. 5. Still, UDOO manages to process $2-3MP$ side-scan sonar images from our real dataset in around $5-10s$ which is fast enough for mission replanning purposes. It is also interesting to analyze processing time in a relative sense normalized by the size of an image in MPs. Fig. 5 shows that, on average, UDOO processes $1MP$ in $1.5-2s$. This can be used as a mission parameter. Knowing the expected size of the sonar image per one track of the lawnmower survey mission and the time needed to process $1MP$, mission operator can decide to cut the sonar image of the whole line in smaller subregions in order to maintain the online sonar image processing and consequential mission (re)planning in case something interesting appears in sonar data.

CPU usage on UDOO when anomaly-detector node enters the $getAnomalyMap$ method (see Section 3, Algorithm 2) is at most $90-100\%$ but only on one core of UDOO's CPU, leaving the other core unblocked for other operations. Memory usage is primarily affected by the depth n_{scales} of the filter bank, defined Section 3. In the tests, the average size of an image was $10MB$. Anomaly detector node used on average $(n_{scales}+1)\times 10MB$. UDOO has $1GB$ of RAM and $1GB$ of swap memory, and since effective filter depth for anomaly detection was empirically proven to be $n_{scales}<4$, this is enough RAM for such applications.

7. CONCLUSION

An AUV or a multi-AUV systems performing autonomous seafloor exploration missions need to perceive their environment in order to be able to replan the mission based on sensor data. For large-scale areas of seafloor exploration one of the most often used sensors is side-scan sonar. Stacked pings of side-scan sonar can be represented as a grayscale image. In this paper the authors implemented and tested five different anomaly/saliency detection methods. Two of the chosen methods were ever applied to side-scan sonar images, while the other three were mostly applied to natural images. Benchmarking of precision-recall performance of the above mentioned methods was performed on a dataset of 1500 simulated side-scan sonar images. The anomaly detection method from Kaeli (2016) had the best precision and recall performance on simulated

sonar images dataset. Its performance was then validated on a dataset of 350 real side-scan sonar images on which it also had good recall-precision performance. The anomaly detection method from Kaeli (2016) was then implemented using OpenCV library and integrated into Robot Operating System (ROS) environment in order to benchmark its processing time on computer onboard AUV Lupis. It turned out to be fast enough to process $1MP$ of sonar image in $1.5 - 2s$ which the authors deem fast enough for large volumes of sonar data.

REFERENCES

Bore, N., Torroba, I., and Özkahraman, Ö. (2019). *SMaRC project - Side-scan sonar simulator*. URL {https://github.com/smarc-project/smarc_simulations}.

Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3), 15:1–15:58.

Chapple, P. (2008). Automated detection and classification in high-resolution sonar imagery for autonomous underwater vehicle operations executive. Australian Government, Department of Defence, Defence Science and Technology Organisation, Maritime Operations Division.

Goldman, A. and Cohen, I. (2004). Anomaly detection based on an iterative local statistics approach. In *2004 23rd IEEE Convention of Electrical and Electronics Engineers in Israel*, 440–443.

Harel, J. (2019 (accessed February, 2019)). *Simple Itti-Koch Saliency Method*. URL {http://www.vision.caltech.edu/~harel/share/gbvs.php}.

Harel, J., Koch, C., and Perona, P. (2006). Graph-based visual saliency. In *Proceedings of the 19th International Conference on Neural Information Processing Systems*, NIPS'06, 545–552. MIT Press, Cambridge, MA, USA.

Hou, X., Harel, J., and Koch, C. (2012). Image signature: Highlighting sparse salient regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(1).

Huang, X., Shen, C., Boix, X., and Zhao, Q. (2015). Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, 262–270.

Itti, L., Koch, C., and Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11), 1254–1259.

Ji, Y., Zhang, H., and Wu, Q.J. (2018). Salient object detection via multi-scale attention CNN. *Neurocomputing*, 322, 130 – 140.

Kaeli, J.W. (2016). Real-time anomaly detection in side-scan sonar imagery for adaptive AUV missions. In *2016 IEEE/OES Autonomous Underwater Vehicles AUV*, 85–89.

Kapetanović, N., Mišković, N., and Tahirović, A. (2019). Side-scan sonar data-driven coverage path planning: A comparison of approaches. In *MTS/IEEE OCEANS '19 Marseille Conference and Exhibit*, 1–6.

Kapetanović, N., Bibuli, M., Mišković, N., and Caccia, M. (2017). Real-time model predictive line following control for underactuated marine vehicles. *IFAC-PapersOnLine*, 50(1), 12374–12379.

Liu, Y., Chen, L., Wang, H., Jiang, L., Zhang, Y., Zhao, J., Wang, D., Zhao, Y., and Song, Y. (2014). An improved differential box-counting method to estimate fractal dimensions of gray-level images. *Journal of Visual Communication and Image Representation*, 25(5), 1102 – 1111.

Ludvigsen, M., Johnsen, G., Sørensen, A.J., Lågstad, P.A., and Ødegård, Ø. (2014). Scientific operations combining ROV and AUV in the Trondheim Fjord. *Marine Technology Society Journal*, 48(2), 59–71.

Manhães, M.M.M., Scherer, S.A., Voss, M., Douat, L.R., and Rauschenbach, T. (2016). UUV simulator: A Gazebo-based package for underwater intervention and multi-robot simulation. In *OCEANS 2016 MTS/IEEE Monterey*, 1–8.

Mishne, G. and Cohen, I. (2013). Multiscale anomaly detection using diffusion maps. *IEEE Journal of Selected Topics in Signal Processing*, 7(1), 111–123.

Mishne, G., Talmon, R., and Cohen, I. (2015). Graph-based supervised automatic target detection. *IEEE Transactions on Geoscience and Remote Sensing*, 53(5).

Noiboar, A. and Cohen, I. (2007). Anomaly detection based on wavelet domain GARCH random field modeling. *IEEE Transactions on Geoscience and Remote Sensing*, 45(5), 1361–1373.

Pizer, S.M., Johnston, R.E., Ericksen, J.P., Yankaskas, B.C., and Muller, K.E. (1990). Contrast-limited adaptive histogram equalization: speed and effectiveness. In *[1990] Proceedings of the First Conference on Visualization in Biomedical Computing*, 337–345.

Plets, R., Dix, J., and Bates, R. (2013). Marine geophysics data acquisition, processing and interpretation. *English Heritage*.

Reed, S., Petillot, Y., and Bell, J. (2003). An automatic approach to the detection and extraction of mine features in sidescan sonar. *IEEE Journal of Oceanic Engineering*, 28(1), 90–105.

Sharma, P. (2015). Evaluating visual saliency algorithms: Past, present andfuture. *Journal of Imaging Science and Technology*, 59(5), 50501–1–50501–17.

Sokolova, M. and Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing and Management*, 45(4), 427 – 437.

Zhai, Y. and Shah, M. (2006). Visual attention detection in video sequences using spatiotemporal cues. In *Proceedings of the 14th ACM International Conference on Multimedia*, MM '06, 815–824. ACM, New York, NY, USA.

Zhu, B., Wang, X., Chu, Z., Yang, Y., and Shi, J. (2019). Active learning for recognition of shipwreck target in side-scan sonar image. *Remote Sensing*, 11(3).

Zhu, P., Isaacs, J., Fu, B., and Ferrari, S. (2017). Deep learning feature extraction for target recognition and classification in underwater sonar images. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 2724–2731.