2nd International Conference on Advances in Signal Processing and Artificial Intelligence (ASPAI' 2020), 1-3 April 2020, Berlin, Germany

Oral Topic: *Applied Artificial Intelligence*

Using Deep Learning for Sonar Targets Localization

Q. Bruel [1], F. Heitzmann [2], D. Morche [2], J. Huillery [3], E. Blanco [3], L. Bako [3]

[1] Univ. Grenoble Alpes, CEA, List, 17 Avenue des Martyrs, F-38000, Grenoble, France
[2] Univ. Grenoble Alpes, CEA, Leti, 17 Avenue des Martyrs, F-38000, Grenoble, France
[3] Ecole Centrale de Lyon, Laboratoire Ampère, 36 Avenue Guy de Collongue, 69134, Ecully, France
Tel.: +33 4 38 78 97 69
E-mail: quentin.bruel@cea.fr

Summary: This paper addresses the problem of target localization in sonar signal in a 2D (range-azimuth) scene. The aim is to propose an approach based on an artificial neural network that outputs a binary occupancy grid. A dataset is generated using a sonar simulator and used to train and validate a deep neural network based on a U-net architecture. A pre-processing chain converts analog data to a form that can be passed through the neural network, in this case a (range-azimuth) 2D map with power received. Finally, the performances of the network are compared to those of an approach built around on a CFAR-based range estimation and a MUSIC-based direction of arrival estimation. The results show that the network is able to provide at least similar performances than the reference approach, without the algorithmic calibration currently required by the latter.

Keywords: Sonar, Deep Learning, Mapping, Detection, Occupancy grid

1. Introduction

When it comes to autonomous driving and robotic navigation in general, knowing where the agents can move without touching an obstacle is a crucial matter. This information can be displayed as a 2D occupancy grid (OG) of the environment indicating the probability for a region of space to be free or occupied (i.e. containing an obstacle). An example is provided in fig. 1. These grids – here called ”probabilistic” – are popular due to their ability to quantify uncertainty and the possibility to fuse sensor measurements [1][2]. The sensors whose measurements are used to build OG can be separated into two categories according to their field-of-view (FOV), i.e. the area of space where they are able to detect obstacles. A sensor is considered to have either a narrow FOV, that we model as a line, or a large one. In this case a large part of the environment is seen, that we can model as being between two boundaries forming a cone. These notions are illustrated in fig. 2.

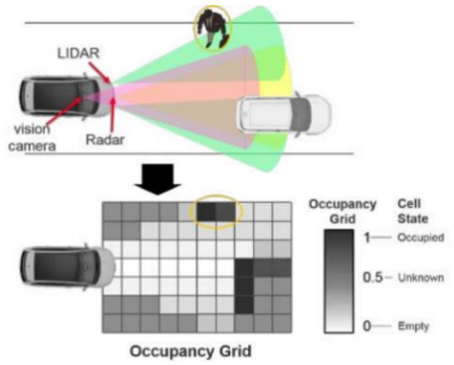


Fig. 1 *Building an occupancy grid from sensor measurements in an automotive context* [1]

If both explicit and efficient methods already allow to compute OG for narrow FOV sensors[1], problems arise when the FOV is large [3]. It is a problem since sensors with large FOV can bring significant advances in autonomous driving and robotic navigation. A first advantage is the larger area covered by a sensor measurement. Another advantage is the fact that the main sensors having a narrow FOV are Lidar ones, and suffer from a heavy cost and inefficiency in scenes presenting optical occlusion. Radar can be seen as a potential answer [4], as well as sonar in certain scenarios. These large FOV sensors present several similarities in the principles behind their signal processing techniques [5]. For these reasons, alongside others, multimodal sensor fusion is seen as a key component for autonomous driving and efficient methods computing OG from large FOV sensors are needed. The existing methods, based on bayesian filtering, are currently either too costly to compute or relying on restrictive hypothesis. To deal with these latter scenarios, the use of deep neural networks has been recently proposed in the literature [6]. Neural networks may provide a suitable approximation of an OG computation model at a reasonable cost.

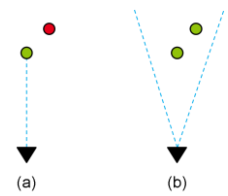


Fig. 2 *An illustration of the difference between narrow (left) and large (right) field of view sensors. The blue dots represent the sensor's field of view. The circles are targets and the ones in green are seen by the sensor.*

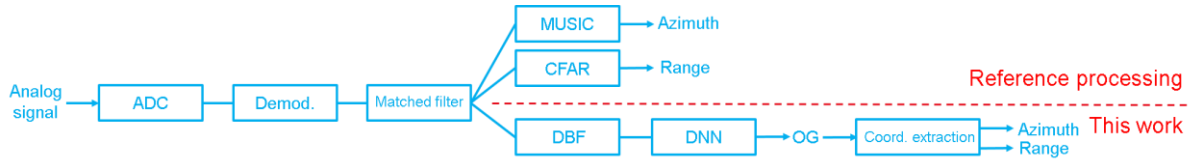


Fig. 3 *A typical sonar signal processing chain and the one used in this work*

Taking benefit from an analogy between 2D range-azimuth maps, OG and images, networks initially intended for image segmentation problems were used for building OG. To do so, the previous works as in [6] use U-net architectures [7] for this type of task. These networks convert the output of the sensor signal processing chain into an OG. The original work used a radar that has a beam too narrow to match our work, even though it produced probabilistic OG. Others have been applied to radar post target detection data (the range and azimuth, as depicted in fig. 3) with ternary OG (the cell can be "free", "occupied" but also "unknown") [8][9].

All these works use metrics that do not give a clear indication of the accuracy of the targets localization. This metric, the Intersection-over-Union (IoU) is the reference in image segmentation. Yet it does not give a direct estimation of the localization accuracy. On top of that, these approaches use "high level" data that have gone through several signal processing algorithms (like those given in fig. 3 as references) that already realize localization tasks. These algorithms have limitations like the need of a complex calibration for the CFAR used as reference [5][10]. Other references of radar target localization using deep learning exist. They addressed this problem in terms of accuracy for the range, azimuth and elevation estimations[11]. However, they do not output occupancy grids.

This work is a continuation of these approaches [6][8][9], using a sonar as a large FOV sensor. A neural network has been trained to generate an OG from sonar signal before target detection. As a preliminary work, the OG only contains binary states: occupied or free. In the mono-target case, accuracy of the target's range and azimuth estimates is evaluated and compared with the estimation accuracy of a reference approach, based on MUSIC and CFAR [10].

2. Reference sonar signal processing chain

This work aims to evaluate the precision of the target localization performed by the network in a 2D context. A target has two main characteristics : its range in the sensor centered referential (in this work in cm) and its azimuthal orientation (in degrees). The comparison is limited to mono-target scenarii. This avoids problems such as occlusion that would complicate the evaluation. Concerning the approach used as reference, the range is obtained using a CFAR detection [10] and time-of-flight calculation. An Order Statistic (OS) CFAR [10], a common version of the algorithm, is implemented and calibrated. This algorithm is the reference when it comes to target detection in radar and sonar, but it suffers some flaws. One of them is the range resolution: if two targets are too closed, it is not possible to separate them. Another is the complicated calibration of the algorithm which is to be used in a specific configuration of noise. As for the direction of arrival (DOA) estimation, the MUSIC algorithm is chosen due to its high resolution [10]. Note that MUSIC needs a priori knowledge of the number of targets present in the scene. The goal is to compare the network's performances to those of the reference. Since the network products OG as output, the range and azimuth are extracted from this OG using image processing tools as depicted in fig. 3. For this purpose, a binary OG is sufficient.

3. Deep Learning based processing chain

The network's architecture is described in this section, followed by the dataset generation.

3.1 Neural network architecture

Similarly with [6][8][9], the network's architecture is a U-net due to its performances in general image segmentation problems. The loss used is the standard binary crossentropy. The network, depicted in fig. 4, acts as an encoder-decoder. A first half extracts features from the input signal and the second converts the feature maps extracted into a signal with the same dimensions as the input one. More specifically, the network is composed of 5 "levels" – not to be confused with "layers" – that can be represented so that it has the "U" shape that gives it its name. The four first levels are present in both encoder and decoder halfs while the 5th is the feature map. An encoder block, as well as the feature map, is composed of two convolutions layers and a max pooling. The same goes for the decoder except that it also comprises a concatenation layer connected to the output of the last convolution layer of

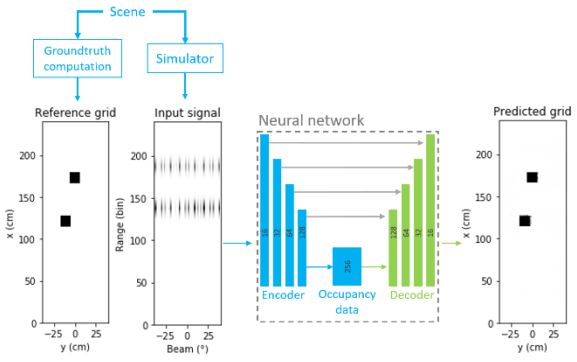


Fig. 4 *The workflow of the DL-based approach. Each rectangle represents 2 convolution layers and a max pooling one. The numbers of convolution kernels for each layer are given in the corresponding rectangles.*

the encoder block in the same level in order to "remap" the features at the right pixels. This way, each pixel of the output image is a cell of the grid associated to the class "free" or "occupied".

3.2 Dataset generation

The perception task that this work is tackling consists in converting a sonar signal into a representation of the environment in front of the sensor. The training of the network needs an input and the output that the network is supposed to produce given such input. It is a supervised learning handling an image segmentation problem [12]. Considering this, the dataset must consist of a sonar signal and the occupancy grid of the environment that induced this signal as groundtruth. It is the output that we expect the network to produce given this sonar signal.

A dataset has to be generated since there is no –to the best of our knowledge- public dataset suitable for the problem addressed in this paper. Automotive datasets already tend to rarely contain radar data [13]. Existing datasets containing radar data are however not fit for this work. The main problem is that the data available only contain signals that have passed through a CFAR detector [14]. This would not allow to test the potential of replacing the algorithm with a neural network. Considering the need of fine tuning the scenes for an evaluation accuracy purpose, simulation is chosen for providing a sufficient dataset with a reasonable cost.

A sonar simulator is used to compute the signal received by the sensor after propagation within a scene containing arbitrarily positioned obstacles. In order to focus on the localization and since it is to be performed on an OG, the targets are as large as a cell and placed so that they perfectly fit into one. The region of interest (ROI) is the 80x240cm rectangle in front of the sensor (with it at the center of the x-axis). This rectangle is partitioned into a grid with the same dimensions (each cell is a 1cm square). A scene consists of between one and three 1cm wide planes. The planes are randomly placed on cells considering a margin of 10 cells on the x-axis and between 35 and 220 cm. 3000 scenes are generated. A problem arises with groundtruth grids computation. It is difficult for a neural network to output an image with a segmentation of only one pixel. Thus, the targets on the groundtruths are not represented as a single cell but rather as a 7x7 square as shown in fig. 5. The square's center of mass represents the coordinates of the detected target.

A sonar is then simulated in order to obtain an input signal to the network. The sensor consists of one emitter so that the emission beam is the widest possible. It also has 5 transducers for reception that allow direction of arrival (DOA) estimation for the echoes. This way, it is possible to estimate the azimuth of the detected targets. The sonar emits a sine pulse and the simulator computes the acoustic wave received at each transducer after reflection on target(s). The signal is amplified and converted into the digital domain. The common processing to both reference and deep learning approaches includes a signal demodulation followed by a matched filter. Since a 2D signal is required, a Digital Beamforming (DBF) algorithm is used for obtaining a 2D polar map representing the power received from each point of space in front of the sensor [15]. The process is illustrated in fig. 3. An example of input signal is also depicted in fig. 4.

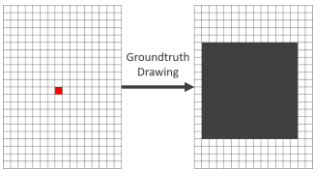


Fig. 5 *In order to provide a groundtruth that allows to train the network, we represent each cell by a bigger square.*

4. Simulation results

Both the reference and neural network's implementations were evaluated in terms of range and azimuth accuracy using the Root Mean Squared Error (RMSE) as a metric:

$$RMSE(y,\tilde{y}) = \sqrt{\frac{\sum_{i=1}^{N}(y_i-\tilde{y}_i)^2}{N}} \qquad (1)$$

Where y_i is the groundtruth and $\tilde{y}_i$ is the value predicted by the model for a validation test of N samples for vectors y and $\tilde{y}$. Since this metric is used for evaluating estimation, we do not count the cases where the target is not detected or where a non-existing target is, which can happen when the SNR drops too low. Results obtained for each cell on the left half of the ROI (miss detections not taken into account) are displayed in table 1.

Table 1. *Performances of both approaches on the validation data (RMSE)*

Approach	Range (cm)	Azimuth (°)
CFAR + MUSIC	4,463	0.424
Neural network	**0.820**	**1.465**

4.1 Results analysis

The network achieved a better accuracy than the reference approach for the range estimation. On the contrary, MUSIC achieved a better result on azimuth estimation, even though the network provided encouraging results. An example of scene reconstruction (in a multi-target scenario) is given in fig. 4. In order to obtain a richer information than the global performance displayed in table 1, a set of 20 simulations was run where the error is evaluated at every possible cell of the grid (this time, a miss detection gives a maximum error). Thanks to the symmetrical RMSE distribution over the x-axis, only the left half of the map was computed. It allows to draw an image containing in each pixel the RMSE for each corresponding cell in the grid. For each approach, an error map is drawn for each of the two estimated

parameters. The resulting maps are displayed in fig. 6.

As expected, the reference approach has trouble detecting targets that are at the extremities of the emission diagram, where it ensues failures. The phenomenom is less pronounced with the neural network, which suffers less miss detections in this area. This aside, the reference approach performs a pinpoint accuracy regarding the azimuth estimation on the rest of the region. The network has trouble with the extremities of the ROI, where the SNR drops. Oberving the grids computed for targets in this area show that the network does not draw perfect 7x7 squares. This adds another source or error since the center of mass of the square does not correspond to the target location. This is not observed for range estimation, where the network keeps giving better results than CFAR and shows a pretty uniform error.

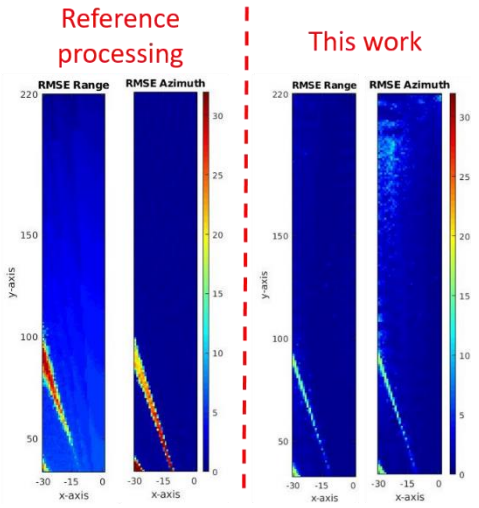


Fig. 6 *The errors observed for each approach estimations of range and azimuth. The errors above 12 are caused by more or less frequent miss detections.*

4.2 Discussion

These results must be regarded in the light of limitations in the possibilities offered by the simulator. The absence of reflective elements besides the targets clearly eases their localization and cannot be considered "realistic". In a concrete sonar situation a lot of constraints such as clutter would considerably complicate the problem and thus reduce the implementations performances [5]. Nevertheless, this work aimed to provide a proof-of-concept for computing occupancy grids from low level sonar data with satisfying performances, which the results tend to infer. These results must now be consolidated by conducting experiments on real world data or more complex simulation scenarios.

5. Conclusions

A deep neural network converting preprocessed sonar data into occupancy grids indicating the location of the targets has been successfully trained. To do so, simulated scenes and associated sonar outputs were used for training and testing the artificial neural network. This network has shown encouraging results for range and azimuth estimation on single target scenario. It holds the comparison with a CFAR and MUSIC combination. Future works shall focus on the robustness of this approach to more challenging conditions in terms of noise, by using real world data. Alongside a certain consolidation, it would also allow to study if it keeps performing well in situations where CFAR and MUSIC show their limitations. It is also intended to work on the generation of probabilistic occupancy grids for sensor fusion purposes.

References

[1]. T. Rakotovao Andriamahefa, "Integer Occupancy Grids : a probabilistic multi-sensor fusion framework for embedded perception", *Ph.D. thesis, Université Grenoble Alpes*, CEA Grenoble, Feb. 2017.

[2]. S. Thrun, W. Burgard, et D. Fox, *Probabilistic Robotics*. MIT Press, 2005.

[3]. R. Dia, "General Inverse Sensor Model for Probabilistic Occupancy Grid Maps Using Multi-Target Sensors," *23rd International Symposium on Mathematical Theory of Networks and Systems*, Hong Kong, p. 8, 2018.

[4]. I. Bilik, O. Longman, S. Villeval and J. Tabrikian, "The Rise of Radar for Autonomous Vehicles: Signal Processing Solutions and Future Research Directions," in *IEEE Signal Processing Magazine*, vol. 36, no. 5, pp. 20-31, Sept. 2019.

[5]. François LeChevalier, Principles of Radar and Sonar Signal Processing, *Artech House Publishers*, 2002.

[6]. R. Weston, S. Cen, P. Newman and I. Posner, "Probably Unknown: Deep Inverse Sensor Modelling Radar," *2019 International Conference on Robotics and Automation (ICRA)*, Montreal, QC, Canada, 2019, pp. 5446-5452.

[7]. O. Ronneberger, P. Fischer, et T. Brox, « U-Net: Convolutional Networks for Biomedical Image Segmentation », *International Conference on Medical image computing and computer-assisted intervention* (pp. 234-241), Springer, 2015.

[8]. L. Sless et al., « Self Supervised Occupancy Grid Learning from Sparse Radar for Autonomous Driving », *arXiv:1904.00415 [cs]*, mar. 2019.

[9]. D. Bauer, L. Kuhnert, et L. Eckstein, « Deep, spatially coherent Inverse Sensor Models with Uncertainty Incorporation using the evidential Framework », *IEEE Intelligent Vehicles Symposium (IV)*, 2019.

[10]. M. I. Skolnik, *Radar Handbook, Third Edition*. McGraw-Hill Education, 2008.

[11]. D. Brodeski, I. Bilik, et R. Giryes, « Deep Radar Detector », in *2019 IEEE Radar Conference (RadarConf)*, 2019, p. 1-6.

[12]. J. Long, E. Shelhamer, et T. Darrell, « Fully Convolutional Networks for Semantic Segmentation », *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431-3440), 2015.

[13]. D. Feng *et al.*, « Deep Multi-modal Object Detection and Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges», *arXiv:1902.07830 [cs]*, feb. 2019.

[14]. H. Caesar *et al.*, « nuScenes: A multimodal dataset for autonomous driving », *arXiv:1903.11027*, mar. 2019.

[15]. B. D. V. Veen et K. M. Buckley, « Beamforming: a versatile approach to spatial filtering », *IEEE ASSP Magazine*, vol. 5, nº 2, p. 4-24, apr. 1988.