



Seabed prediction from airborne topo-bathymetric lidar point cloud using machine learning approaches

Julian Le Deunf, Rudresh Mishra, Yves Pastol, Romain Billot, Steve Oudot

► To cite this version:

Julian Le Deunf, Rudresh Mishra, Yves Pastol, Romain Billot, Steve Oudot. Seabed prediction from airborne topo-bathymetric lidar point cloud using machine learning approaches. OCEANS 2021: San Diego – Porto, Sep 2021, San Diego, United States. pp.1-9, 10.23919/OCEANS44145.2021.9706113 . hal-03583749

HAL Id: hal-03583749

<https://imt.hal.science/hal-03583749>

Submitted on 22 Feb 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Seabed prediction from airborne topo-bathymetric lidar point cloud using machine learning approaches

Julian Le Deunf
Development bathymetry department
Shom
Brest, France
<https://orcid.org/0000-0003-3644-5063>

Rudresh Mishra
Research Engineer
INRIA Saclay
Paris, France
rudresh.mishra@outlook.com

Yves Pastol
Coastal altimetry department
Shom
Brest, France
yves.pastol@shom.fr

Romain Billot
IMT Atlantique
Lab-STICC, UMR CNRS 6285
Brest F-29238, France
romain.billot@imt-atlantique.fr

Steve Oudot
DataShape team
INRIA Saclay
Palaiseau, France
steve.oudot@inria.fr

Abstract—Predicting the seabed from unfiltered bathymetric lidar data is a very complex task and a critical issue in bathymetric data processing especially with the objective of nautical charting. This is challenging to ensure a high level of quality and security for the needs of a national hydrographic office. This paper proposes a methodology to predict the seabed based on machine learning, which could be useful to automate outlier detection and control the topo-bathymetric lidar point cloud datasets. Several predictive methods have been investigated to predict the seabed from our 2D + 1D data structure. A characteristic dataset of Corsica region was used as a case study for this predictive workflow.

Index Terms—data processing, topo-bathymetric lidar data, supervised machine learning methods

I. INTRODUCTION

Bathymetric point cloud processing is a critical task for the elaboration of nautical charts and its automation is challenging. In particular, the processing of topo-bathymetric lidar data is extremely complex, and is still done completely manually today by the hydrographers and operators of Shom (French Naval Hydrographic and Oceanographic Service). Indeed, the final objective is to plot bathymetric information on nautical charts, which have a legal status. These data are also critical to assess the impact of erosion, flooding risk and the effects of climate change along the coast. They provide key informations for the evolution of ecosystems and underwater habitats and the impact of past or planned coastal protection works.

Figure 1 shows the initial errors in a topo-bathymetric lidar dataset and the result after manual processing. In this context, deciding whether or not soundings have to be invalidated only relies on the decision of a trained hydrographer [1]. Even if this task is supported by dedicated visualization software for a detailed inspection of all soundings, the process is tedious, time-consuming and does not guarantee that all bathymetric features that could rise a navigational risk have been preserved (potential subjectivity of this type of processing). The objective

of our research is to estimate the real seabed from topo-bathymetric lidar point cloud in order to provide operators with a tool for processing and quality checking bathymetric data.

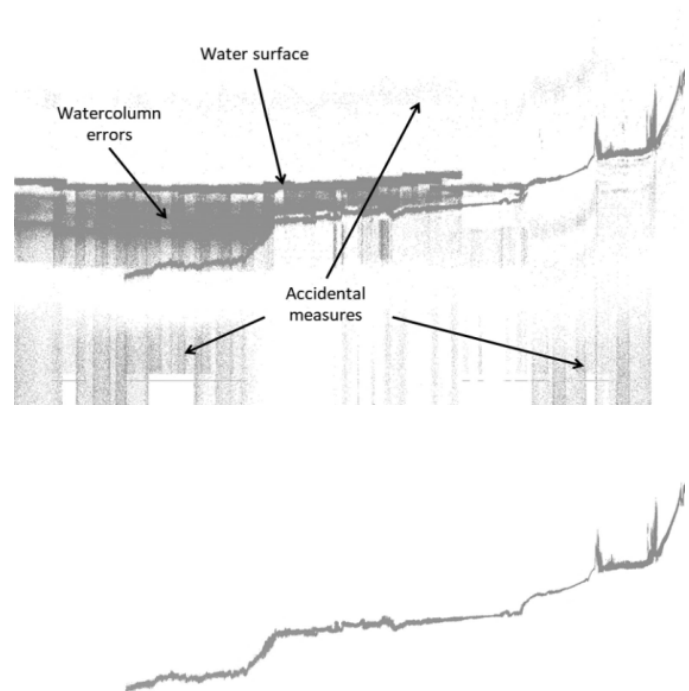


Fig. 1. Types of errors depicted through a bathymetric section (top), sea bottom and topography after manual processing (bottom)

The remaining of this article is organized as follows. In Section II we briefly introduce the state of the art concerning

outlier detection in topo-bathymetric lidar data, and we present its counterpart for Multibeam Echo Sounder (MBES) data, which is very close in terms of characteristics and structure of the acquired datasets. Then, in Section III we explain our approach and especially the data structure we have built, before presenting in Section IV a practical case study from Shom's Corsica data. We then discuss in Section V some of our conclusions and present future perspectives raised by this work.

II. METHODOLOGICAL BACKGROUND

The state of the art about point cloud manipulation and processing is abundant, especially with the contributions of recent deep learning research, as shown in the paper [2] presenting a new point cloud segmentation method, PointCNN, and its performance compared to other classical methods, such as PointNet or SPGraph. These rapid advances in the field of deep learning for lidar data are linked to the rise of autonomous vehicles and the use of this technology for their navigation, as shown in [3].

A. Topo-bathymetric lidar data

The recent evolution of bathymetric lidars, improving the acquisition capacities (for example by increasing the density) favors the use of methods associated with machine learning in the context of processing bathymetric point cloud, as shown in two recent papers [4, 5] dealing with the classification of topo-bathymetric lidar point cloud.

The authors of [4] investigate the classification of a lidar point cloud, from an AHAB Chiroptera I, acquired in September 2013. The authors propose to build a set of 15 features based on the intensity measurements of the lidar sensor and on the local geometry of the point cloud (by building a cylindrical neighborhood around the studied soundings). These descriptors are then normalized (standard score) and integrated into an artificial neural network with 2 hidden layers (of respectively 15 and 7 neurons). The terminal layer provides the three classes (seabed, seabed object and water surface) that the method is aiming to predict. Considering these results, the real problem seems to be the ability to detect objects on the bottom (the accuracy of the classification of the seabed or the water surface is always above 98.5%).

The authors of [5] also seek to classify (binary classification: bathymetric or non-bathymetric data) a bathymetric lidar point cloud based on descriptors built from the raw data characteristics (such as number and return number, position or attitude measurement uncertainty, incidence angle, etc.). The authors used 14 descriptors classified in three categories (return-based, SBET and lidar-edge). The test data was acquired by NOAA via a RIEGL VQ-880G off the coast of Florida in April 2016 and unlike the previous paper the data volume is much larger (983320 soundings processed). The authors worked with 3 supervised classification methods: regularized logistic regression, multi-layer perceptron neural networks and regularized extreme gradient boosting (XGBoost). Nevertheless, only the results of the last model are presented. The confusion matrix

indicates a very good accuracy (99.6%) but this is nuanced by the imbalance class found in the input dataset.

These two articles adopt the same standpoint: trying to classify a point cloud from features intrinsic to the data or considering an auxiliary factor (e.g spatial information or strength of the signal return). The prediction is also always done on the point (no data rasterization) even if a neighborhood can be built locally during features' generation. Furthermore, the metrics used are very global and do not ensure that local artefacts or slope breaks have been properly classified.

B. MBES data

Here a parallel is made with the MBES data because hydrographers were very early interested in the automation of the bathymetric point cloud processing. Indeed, since the arrival of MBES sounders in the 1980s, numerous techniques for detecting outliers in bathymetric point cloud have been proposed, each with very specific objectives (detection of underwater pipelines, generation of a global model, detection of point or group outliers, etc.). These different techniques have been described and compared in the literature review presented in [6].

Indeed the authors of [6] present an exhaustive taxonomy of MBES data processing algorithms from the 1980s to the present. This classification shows the imbalance between methods relying almost exclusively on unsupervised learning (33 papers selected) and a method [7] relying on a supervised technique. The classification then shows two different families of processing methodologies: the data-oriented (methods centered on a sounding and its spatial or temporal neighborhood), and the surface-oriented (methods centered on the generation of a surface supporting the (semi)-automatic processing). In that paper, the authors also present the advantages and disadvantages of these different methods.

The surface-oriented approaches, close to what we want to implement in terms of algorithmic output, often have the same limitations: a tendency to level the seabed bottom and remove the shallow items (e.g. wrecks, obstruction). However, these bathymetric artefacts are crucial for a national hydrographic office which seeks to preserve all the seabed structures in order to integrate them into nautical products. Hence our objective is to take advantage of the high density offered by the lidar data in shallow water to use supervised methods associated with these surface-oriented approaches. In [7], the authors propose to classify bathymetric point cloud from MBES sensors based on a voxel data structure. Again, a binary classification has been performed between the data to be kept and the data considered as outliers. The model used for this prediction is adapted from the UNet neural network [8], classically used in medical imaging segmentation. This architecture being based on the principle of auto-encoders, no features are built *a priori*.

C. Knowledge gaps and contributions

In contrast to previous bathymetric lidar research (see [4, 5]), which focuses on the classification of bathymetric lidar

point cloud, we have formalised our question as a regression problem. Indeed, in the context of the exploitation of bathymetric lidar point clouds for nautical products, and especially for navigation safety, it is interesting to build a model based on a regression and not a strict binary classification. Moreover, these regressive approaches do not appear to have been used in previous studies. In addition, by generating a digital bathymetric model, the output of the predictive workflow is much easier to understand for the operators, in terms of control and data quality, than a flag (accepted/rejected) on soundings.

The objectives of our study are to make the following contributions:

- Combining a surface-oriented approach with supervised regression methods
- Comparing the performance of different regressors with our approach.
- Building up a new 2D + 1D data structure.
- Using the complementarity of sensors.
- Scaling up with large datasets for learning and testing (over 40 million soundings per dataset).

III. PROPOSED APPROACH

Our approach estimates the seabed by producing a digital bathymetric model from a regression algorithm. We seek to produce a bathymetric surface from a point cloud with supervised algorithms.

This paradigm shift, compared to the papers presented in Section II-A, has two objectives:

- To be able to rely on the research associated with raster data processing, working with classical image processing [9] but also deep learning (such as for hyperspectral data in [10]), which has matured considerably in recent years.
- To allow the operators to compare the generated surface with the point cloud and use this surface as a processing or control tool depending on the complexity of the considered area, thus producing a decision support tool in addition to an automatic workflow.

A. Data structure and features

To perform this regression, our proposal is based on a data structure depicted in Figure 2, which encodes a 2D (North/East) + 1D (elevation from the ellipsoid) voxel grid in which we store a set of features. This sort of data structure is also presented in the context of classification work based on deep learning on hyperspectral data - see [10]. It allows for resolutions on the vertical and horizontal axes, the density of points being very variable in these two dimensions, making it possible to better take into account the source data.

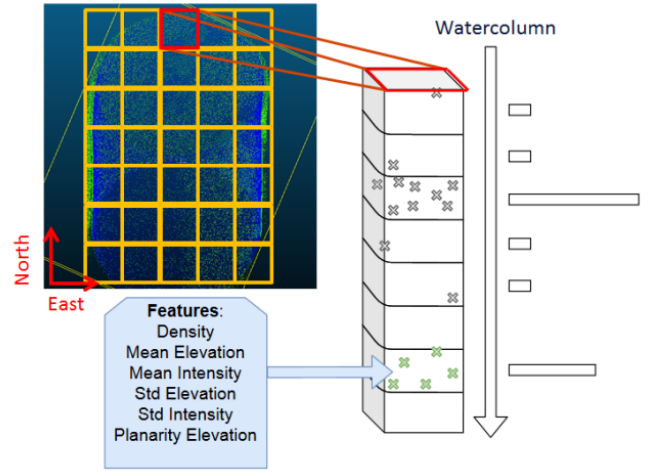


Fig. 2. Data structure and features build for our regressor.

The features currently considered and computed for each voxel composed of n soundings (with elevation z and intensity I) are:

- unnormalized density d of soundings per voxel:

$$d = n \quad (1)$$

- mean elevation \bar{z} :

$$\bar{z} = \frac{1}{n} \sum_{i=1}^n z_i \quad (2)$$

- standard deviation σ_z of elevation:

$$\sigma_z = \sqrt{\frac{1}{n} \sum_{i=1}^n (z_i - \bar{z})^2} \quad (3)$$

- mean of the return signal intensity \bar{I} :

$$\bar{I} = \frac{1}{n} \sum_{i=1}^n I_i \quad (4)$$

- standard deviation σ_I of the return signal intensity:

$$\sigma_I = \sqrt{\frac{1}{n} \sum_{i=1}^n (I_i - \bar{I})^2} \quad (5)$$

- the flattening of the ellipsoid associated with the covariance matrix of the soundings P_λ :

$$P_\lambda = \frac{\lambda_2 - \lambda_3}{\lambda_1} \quad (6)$$

Where λ_1 , λ_2 and λ_3 are the eigenvalues calculated on the point cloud present in the studied voxel. This last feature, flattening, captures the shape of the point cloud and thus allows to describe the voxel roughness (which could be very discriminating between the water surface and the seabed for example).

This data structure feeds a regressor, whose role is to predict the seabed elevation within each watercolumn. The regression

methods we tested so far are: Support Vector Regression (SVR), Random Forests (RF) and MultiLayer Perceptron (MLP), which will be presented in details in the following sub-section.

B. Sampling and modeling

We use a dataset of about 500,000 soundings (a first experimental dataset to tune the models) that have been processed with the features described in the previous sections. In our approach the ground truth is the average elevation of the soundings considered valid by an operator in a horizontal pixel. This ground truth is therefore very unevenly distributed in terms of elevation representation as shown in Figure 3. In order to make the data unbiased and fit the varied ground truth surface level characteristics, a bin histogram sampling method is applied.

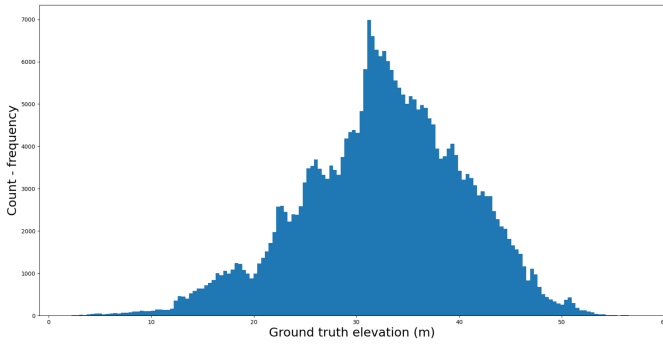


Fig. 3. Elevation histogram for 1st dataset ground truth.

Suppose there are k bins. Let n_i be the count assigned to the i_{th} bin, and let a_i and b_i be its left and right edges. We applied a very basic algorithm to sample the dataset. We computed min_bin_size and $mean_bin_size$ from the overall bin frequency; then c_i , which is the count of randomly sampled data in each bin, is computed using the following equation:

$$c_i = \begin{cases} rand(mean_bin_size) & bin_freq > min_bin_size \\ 0 & bin_freq < min_bin_size \end{cases}$$

We trained three regression models on the bathymetric data: Support Vector Regression (SVR), Random Forests (RF) and MultiLayer Perceptron (MLP).

SVR is considered as a non-parametric technique because it relies on kernel functions. It aims to reduce the error by determining the hyperplane and minimizing the difference between the predicted and observed values. It tries to fit the best line within a threshold value (Distance between hyperplane and boundary line). It uses the points with this boundary to predict the value. For a non-linear regression, the kernel function transforms the data to a higher dimension and performs the linear separation. Here we will use the radial basis function kernel.

RF for nonlinear regression are trained by growing trees dependent on a random vector, so the tree predictor takes numerical values as opposed to class labels. It takes several

(but different) regression decision trees and "votes" for them. The random forest regression then averages all the predictions to generate a better estimation of the expected ground truth.

The architecture of the MLP consists of four layers of neurons, with complete connections between the layers. The first (input) layer contains 128 neurons, the second (intermediate) layer contains 64 neurons, the third layer contains 8 neurons, and the last layer (output) contains 1 neuron. Neurons in the first three layers have a relu activation function of the following form:

$$f(x) = \begin{cases} 0 & x < 0 \\ x & x > 0 \end{cases}$$

The Neuron in the final layer has a linear activation function to perform regression.

By using the linear layer at the output of an MLP, the regression process obtains the predicted value. The model has been trained using the error back propagation algorithm with a learning coefficient of $\rho = 0.01$, the model loss is the mean absolute error, and the optimizer is the Adam optimization algorithm.

For the modeling, different horizontal and vertical resolutions were taken into account in order to consider the different point cloud densities. The trade-off was to try to have the highest possible resolution (in order to better describe the seabed) while keeping enough soundings for the regression to be statistically relevant while considering computation times and memory constraints compatible with the computers at our disposal (96 Go RAM, 1To HDD, 2.50 GHz CPU, NVIDIA Quadro RTX 5000).

The following tables depict the mean absolute error (MAE) for different pixel resolutions and vertical resolutions for the RF, SVR and MLP regressor model.

TABLE I
MEAN ABSOLUTE ERROR (MAE) FOR RF, SVR AND MLP PREDICTION FOR 20 METERS HORIZONTAL RESOLUTION

Regressor	Vertical resolution	MAE
RF	5m	0.92m
SVR	5m	3.13m
MLP	5m	1.36m
RF	2m	0.73m
SVR	2m	3.04m
MLP	2m	1.48m

The results from Tables I to IV show that the MAE loss of the SVR model tends to be larger compared to the RF and MLP models when the vertical resolution is set to 5 meters or 2 meters. For the rest of the analysis, we did not consider the

TABLE II
MAE FOR RF, SVR AND MLP PREDICTION FOR 15 METERS HORIZONTAL
RESOLUTION

Regressor	Vertical resolution	MAE
RF	5m	0.78m
SVR	5m	3.05m
MLP	5m	1.12m
RF	2m	0.62m
SVR	2m	2.99m
MLP	2m	6.79m

TABLE III
MAE FOR RF, SVR AND MLP PREDICTION FOR 10 METERS HORIZONTAL
RESOLUTION

Regressor	Vertical resolution	MAE
RF	5m	0.58m
SVR	5m	2.70m
MLP	5m	1.03m
RF	2m	0.42m
SVR	2m	2.68m
MLP	2m	1.08m

TABLE IV
MAE FOR RF, SVR AND MLP PREDICTION FOR 6 METERS HORIZONTAL
RESOLUTION

Regressor	Vertical resolution	MAE
RF	5m	0.59m
SVR	5m	2.74m
MLP	5m	0.97m
RF	2m	0.38m
SVR	2m	2.73m
MLP	2m	0.77m

SVR model because of its higher error, thus saving computing time for RF and MLP which seem more promising. The SVR method should be retested in the future with more features or different kernels, rather than the radial basis function kernel used here, to ensure that it is indeed less efficient.

Moreover as we wanted to reduce the horizontal resolution to improve the density/fidelity trade-off of the model. Table V shows the results for the various smaller resolutions.

TABLE V
MAE FOR RF AND MLP PREDICTION FOR HIGHER HORIZONTAL &
VERTICAL RESOLUTION

Regressor	Horizontal resolution	Vertical resolution	MAE
RF	5m	1m	0.28m
MLP	5m	1m	0.66m
RF	5m	0.5m	0.38m
MLP	5m	0.5m	0.44m
RF	2m	0.5m	0.57m
MLP	2m	0.5m	0.69m
RF	1m	0.5m	0.44m
MLP	1m	0.5m	1.09m

After several iterations at different resolutions, both models showed optimal results when the horizontal resolution is 5m.

Nevertheless, as the input dataset is small, we have sought to improve these results by taking into account more soundings, and we have focused on the 2m horizontal resolution, which is the most effective compromise found so far with the operators (in terms of seabed morphology and model performance).

IV. CORSICA CASE STUDY

In order to test our approach at a larger scale, we have applied it to a Corsican dataset acquired in September 2018 as part of the Shom's Litto3D® project (see [11]), which aims to produce a complete land-sea reference system for France. These data were acquired onboard a Cessna 208 B Grand Caravan aircraft carrying a Leica HawkEye 3, deploying a triple laser: red laser for topography, shallow green laser for shallow water, and deep green laser for deeper water. The data in Section III-B are derived from this acquisition. The environmental conditions were particularly mild in the area and the lidar was able to reach 40 meters depth.

Since the SVR performance in the preliminary study of Section III-B was disappointing, we did not use it in the Corsica case study.

A. First iteration on a small dataset

In order to test our approach, a first model was built with a small dataset of 15,631,391 soundings (see Figure 4) with topographic, shallow and deep laser (i.e. all lidar sensors available on an area).

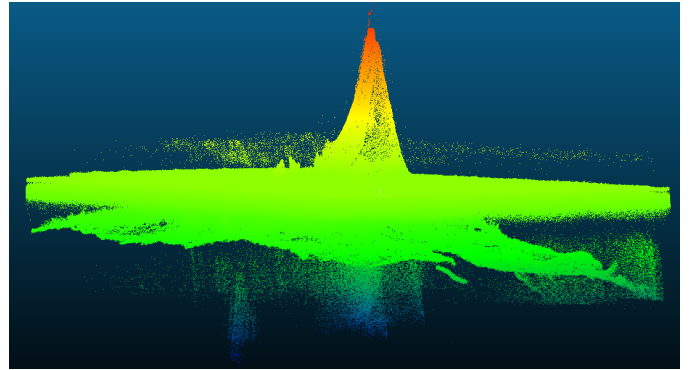


Fig. 4. Raw dataset with all lasers, scaled up by 10 on the Z-axis.

Table VI shows the results obtained on the dataset of Figure 4 with the RF and MLP predictors, with a horizontal resolution of 2m and a vertical resolution of 0.5m. This choice was made in order to have the most accurate morphology possible.

Note that keeping the measurements (and associated features) from all the different sensors in the data structure can make the information saved in the voxels confusing. Figure 5 shows an example of a data subset with the deep laser in red

TABLE VI
MAE AND STANDARD DEVIATION (STD) FOR RF AND MLP

Regressor	MAE	STD
RF	0.28m	1.26m
MLP	0.66m	1.43m

and the shallow laser in green: we can see that there is a significant overlap between the deep and shallow laser data.

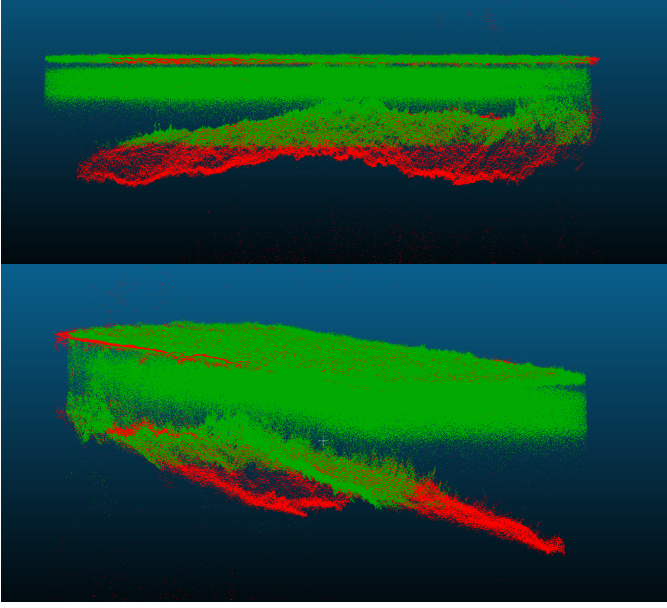


Fig. 5. Side view and perspective view of a subset of data, lidar shallow in green and lidar deep in red.

This overlap between the two sensors can perturb the features construction. Indeed, the points' density in an overlapping area will increase significantly and then decrease drastically when the shallow laser is no longer present. Moreover, the overlap may induce a change in light intensity return, as shown in Figure 6. This change in the intensity of the return signal is explained by the difference in power and signal absorption in the watercolumn for the deep and shallow lasers, see [12].

We have therefore modified our data structure as follows, in order to improve the results of the regression models.

B. Separation of sensors and views

As just mentioned, keeping the different sensors in the same data structure has the effect of diluting the quality of the different features. We have therefore separated the different sources (deep and shallow laser) and kept only the data sources containing bathymetric data (excluding the topographic red laser).

In addition, we have separated the front view from the rear view considering that during the aircraft flight path the acquisition "scenes" were different, which will allow in the future to combine these two views to make the prediction

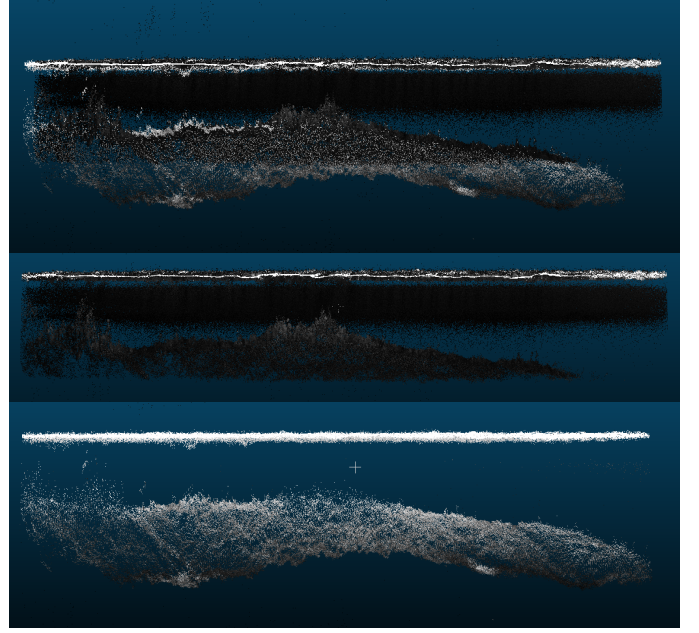


Fig. 6. At top both shallow and deep laser intensity. In the middle only shallow laser intensity. At the bottom deep laser intensity. The grey scale intensity is the same for both sensors.

more robust. Furthermore, in this subsection we will only present the results of the random forest which, as shown in the previous section, provides more accurate results than the MLP. We tested this approach on 4 different datasets (2 flight lines or more (FL) for each dataset) from the same area in Corsica. Table VII presents the results of the RF on these 4 areas. A horizontal resolution of 5m was chosen for the deep laser and 2m for the shallow laser, a vertical resolution of 0.50m was chosen in both cases.

TABLE VII
MAE (IN METERS) AND STD (IN METERS) FOR RF PREDICTION FOR DIFFERENT SENSORS AND VIEW

Sensors and view	FL 528-531	FL 438-439	FL 440-441	FL 442-443
Shallow rear	0.09 / 0.28	0.13 / 0.41	0.11 / 0.34	0.16 / 0.51
Shallow front	0.11 / 0.38	0.13 / 0.40	0.12 / 0.34	0.15 / 0.49
Deep rear	0.26 / 1.33	0.36 / 1.37	0.32 / 1.31	0.21 / 0.99
Deep front	0.22 / 1.34	0.32 / 1.28	0.33 / 1.34	0.24 / 1.32

Results with the lowest MAE and STD for the shallow and deep sensors are shown in bold in Table VII. It is interesting to note that for the shallow sensor the MAE varies between 0.09m and 0.16m, which is comparable to the noise present in the lidar data (between 0.10m and 0.30m for the shallow sensor). The STD varies between 0.26m and 0.51m and appears to be marked by locally significant errors (see Figure 9 for shallow errors). For deep sensors the MAE varies between 0.21m and 0.36m, which is also comparable to the noise present in the lidar data (between 0.30m and 0.50m for

the deep sensor).

C. Shallow laser dataset

Figure 7 shows the FL 528-531 data set for the shallow rear laser. The colorbar represents the elevation value with respect to the GRS80 ellipsoid.

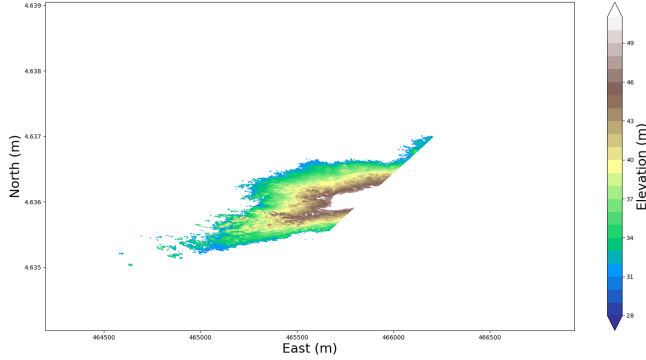


Fig. 7. Original dataset with a 2m pixel grid size for FL 528-531 shallow rear lidar.

Figure 8 represents the sampled pixels used to compute the RF model, based on a 70% train and 30% test split. The colorbar represents the elevation value with respect to the GRS80 ellipsoid. Here the learning conditions are optimal because a random sampling has been carried out. It will be interesting in the future to vary this learning strategy to see how the model is able to generalize.

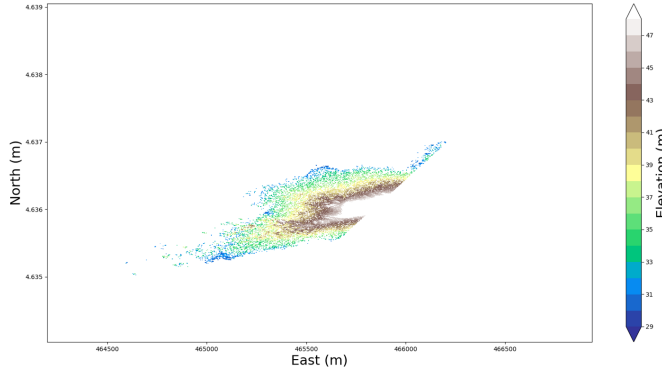


Fig. 8. Training FL 528-531 shallow rear dataset used for RF prediction.

Finally, Figure 9 shows the difference for each pixel between the ground truth and the prediction (only on the test data). It is interesting to note that, as indicated by the MAE in Table VII, the error is globally weak on the whole dataset. There are however some localised errors around areas where small morphological artifacts (compared to the grid pixel size) are present or where few training samples are available (typically in the shallowest areas) or on some boundaries.

D. Deep laser dataset

Figure 10 shows the ground truth of flight lines 528-531, with a horizontal resolution of 5 meters and the deep rear laser.

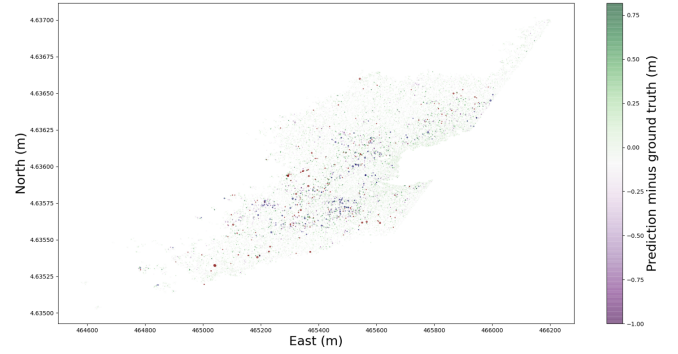


Fig. 9. Difference between the ground truth and prediction with FL 528-531 shallow rear dataset (on test samples only).

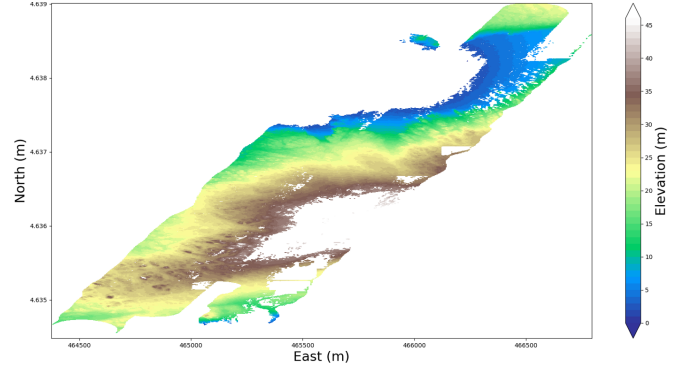


Fig. 10. Original dataset with a 5m pixel grid size for FL 528-531 deep rear dataset.

Figure 11 shows the sampling used for supervised learning with the RF, using the same 70/30 ratio as with shallow laser.

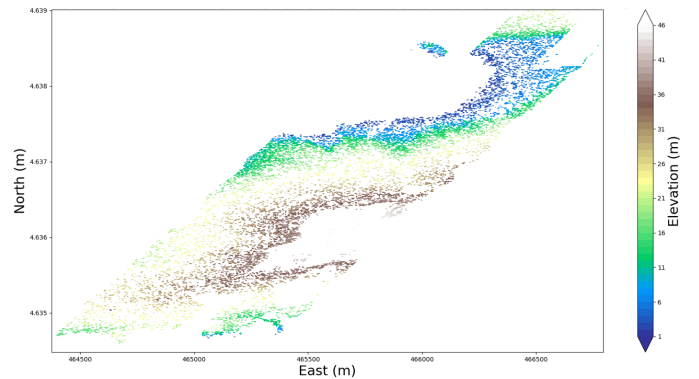


Fig. 11. Training FL 528-531 deep rear dataset used for RF prediction.

Finally, Figure 12 shows the localised error for the deep laser. Here again the error is globally weak on the whole dataset. However, there are again some localised errors around areas where small morphological artifacts (compared to the grid pixel size) are present, or where few training samples are available (typically in the shallowest areas), or close to the boundary of the domain (where the computed features are perturbed by the presence of the boundary).

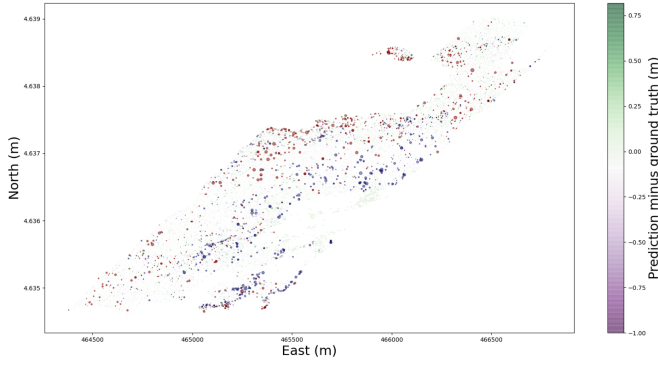


Fig. 12. Difference between the ground truth and prediction for FL 528-531 deep rear dataset (on test samples only).

E. Bias in prediction

Figures 9 and 12 highlight some correlation between the sign of the errors and the predicted depth. Figures 13 and 14 show a negative correlation between the signed prediction error and the elevation in the ground truth. This behavior is not in accordance with the safety of navigation, for which it is important not to underestimate the elevation in shallow areas. This behaviour will be further studied in a next study, in order to understand the origin of the bias.

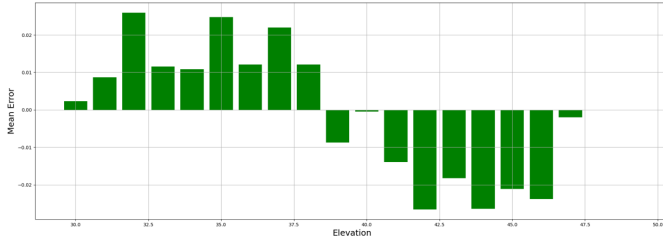


Fig. 13. Mean difference between ground truth and prediction error per elevation range for FL 528-531 shallow rear dataset.

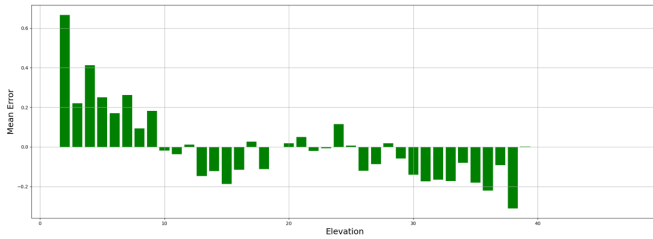


Fig. 14. Mean difference between ground truth and prediction error per elevation range for FL 528-531 deep rear dataset.

V. PERSPECTIVES

In our future work, the selection of the training set will depend on the scenario considered. Indeed, we will consider 2 different scenarios, depending on the existence of the data present in the area – please refer to Figure 15. In the first scenario, operators have already processed a portion of the lidar swath (marked in red), which we use as training data.

In the second scenario, operators have processed the area (or some neighboring area) during a former survey and we can use this history as training data.

Moreover, there is still a lot of work to be done on the descriptors. Indeed, we can add elements to better describe the point cloud (by taking example on the article [4]), but also by building descriptors based on data clustering like DBSCAN [13] or ToMATo [14].

Finally, as we have seen in Section IV, many of the prediction errors are localized in very narrow areas, which suggests that a spatial regularization could be applied in order to improve the final prediction by taking into account the neighboring horizontal pixels. We could also set up combinations of different views and different sensors allowing for multi-source and multi-scale data fusion.

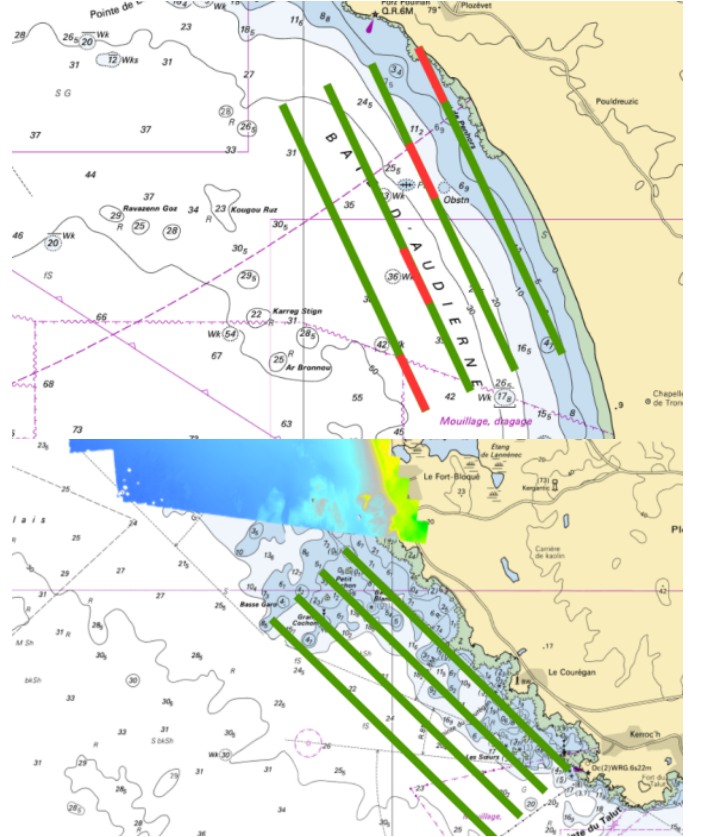


Fig. 15. Different learning scenarios considered. Top: learn after a phase of partial manual processing (in red). Bottom: learn from a previously processed area (represented by the digital elevation model on top).

VI. ACKNOWLEDGMENTS

The authors want to thank the Shom for providing access to the data in this research project as well as access to lidar data experts for feedback. This project is funded by the Direction interministérielle du numérique as part of a call for expressions of interest on artificial intelligence.

REFERENCES

- [1] IHO, “The standards for hydrographic surveys,” tech. rep., IHO, 2020. Special Publication N°44, 6 edition.

- [2] Y. Li, R. Bu, M. Sun, and B. Chen, "Pointcnn," *CoRR*, vol. abs/1801.07791, 2018.
- [3] Y. Li, L. Ma, Z. Zhong, F. Liu, M. A. Chapman, D. Cao, and J. Li, "Deep learning for lidar point clouds in autonomous driving: A review," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, 2020.
- [4] T. Kogut and A. Słowik, "Classification of airborne laser bathymetry data using artificial neural networks," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 1959–1966, 2021.
- [5] K. Lowell, B. Calder, and A. Lyons, "Measuring shallow-water bathymetric signal strength in lidar point attribute data using machine learning," *International Journal of Geographical Information Science*, vol. 35, pp. 1–19, 12 2020.
- [6] J. Le Deunf, N. Debese, T. Schmitt, and R. Billot, "A review of data cleaning approaches in a hydrographic framework with a focus on bathymetric multibeam echosounder datasets," *Geosciences*, vol. 10, no. 7, 2020.
- [7] D. Stephens, A. Smith, T. Redfern, A. Talbot, A. Lessnoff, and K. Dempsey, "Using three dimensional convolutional neural networks for denoising echosounder point cloud data," *Applied Computing and Geosciences*, vol. 5, p. 100016, 2020.
- [8] X.-J. Mao, C. Shen, and Y.-B. Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," in *Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16*, (Red Hook, NY, USA), p. 2810–2818, Curran Associates Inc., 2016.
- [9] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 6, pp. 1397–1409, 2013.
- [10] N. Audebert, B. Le Saux, and S. Lefevre, "Deep learning for classification of hyperspectral data: A comparative review," *IEEE Geoscience and Remote Sensing Magazine*, vol. 7, p. 159–173, Jun 2019.
- [11] Y. Pastol, "Use of Airborne LIDAR Bathymetry for Coastal Hydrographic Surveying: The French Experience," *Journal of Coastal Research*, pp. 6–18, 03 2011.
- [12] W. Philpot and J. Wozencraft, "Airborne Laser Hydrography II," in *American Geophysical Union*, vol. 2016, pp. IS14B–2313, Feb. 2016.
- [13] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD'96*, p. 226–231, AAAI Press, 1996.
- [14] F. Chazal, L. J. Guibas, S. Y. Oudot, and P. Skraba, "Persistence-Based Clustering in Riemannian Manifolds," *Journal of the ACM (JACM)*, vol. 60, p. 38, Nov. 2013.