Ocean Floor Contour Prediction

Merve Kilic
Electrical and Computer Engineering
UC San Diego
La Jolla, CA
`mkilic@ucsd.edu`

Maxime Ghesquiere
Electrical and Computer Engineering
UC San Diego
La Jolla, CA
`mghesqui@ucsd.edu`

Coauthor
Affiliation
Address
`email`

Abstract

...
(Merve)

1 Introduction

1.1 Background

The earth is 70% covered by the ocean and less than 20% has been fully mapped, explored, and observed. This situation is attributed to the same electromagnetic absorption qualities of water that made life on Earth possible, rendering detailed seafloor mapping difficult. Electromagnetic waves in all spectra reach at most 100m depth in the ocean where the average depth is 3688m. [Fig. 1.1] As a result, most satellite and remote sensing techniques are impossible to describe the ocean floor accurately.

To achieve any reasonable level of accuracy, modern bathymetry relies on the Doppler effect of acoustic signals propagating through the water. Because acoustic signals require a sufficiently dense medium to propagate, where the denser the medium the stronger the acoustic return. In effect, the high density of water allows for high-accuracy mapping that is not possible with EM waves. The most modern bathymetry methods rely on Multi Beam Echosounder (MBES) mounted on the bottom of ships, then run in strip mine patterns across unmapped areas. However, the cost of operation is prohibitively expensive and gaps would still remain due to the strip mining pattern. For example, to map a 100km^2 area with 500m tracks at 7.4km/h speed on the Robert E. Sproul [Table 1] would require $(100,000 * 2/7.4 = 2702.07h)$ and in excess of $(2702.07\ 24 * 15,000 = \$1,689,189.19)$. Not accounting for travel time from port to destination and return. For reference, Disney World in Florida is 100km^2.

1.2 Motivation

Hence, the high costs of ocean bathymetry have made full mapping of the ocean floor an almost impossible task. Instead, most ocean floor data available are low-quality contours derived from slight height differences of large underwater structures reflected on the ocean surface. However detailed information such as sand banks, miniature trenches, and wrecks are difficult to estimate due to their small mass. We hypothesize that by using machine learning algorithms it would be possible to fill in the gaps and predict regular trends on the ocean floor.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

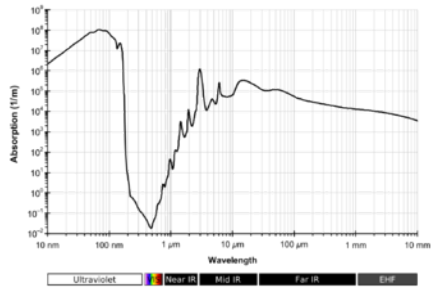

Figure 1: EM wave penetration depth in seawater

Table 1: Typical Research Vessel Costs

Ship name	Price(USD)	Ship Type
Sproul	\$15,000	Regional
Sally Ride	\$45,000	Ocean
Revelle, Meteor, Atalante	\$60,000	Global
Polarstern	\$90.000/day	Ocean

(Mike?)

2 Related Work

(Mike?)

3 Methodology

(state what models we are using which what type of data and our evaluation metrics and why we chose those evaluation metrics)

(Merve)

3.1 Dataset

(describe dataset size and features and how we got it and how we preprocessed it)

(?)

3.1.1 Data Collection

(Mike?)

3.1.2 Data Preprocessing

The collected images have varying height and width, making it hard to pass them directly into machine learning algorithms. In order to standardize our data, we split each image into a grid of 100pix by 100pix squares. Each square is considered a training datapoint. The input to the networks is an 80pix by 80pix center crop, and the ground truth is the full square image.
The data also contains NaN values in locations that have not been surveyed. We deal with these NaNs by replacing them with the mean of the non-NaN values in the ground truth. Some samples have only NaN values, these are thrown out as unusable.

3.2 Models

(explain model architectures and why model was chosen)

(Merve)

3.2.1 Linear Regression

(Merve)

3.2.2 Convolutional Neural Network

(Merve)

3.2.3 Pretrained U-Net

[1]

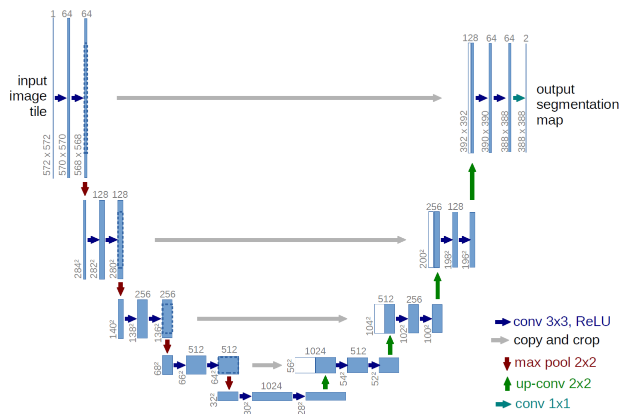

Figure 2: U-Net Architecture [1]

(Merve)

3.2.4 Conditional GAN

Another model we wanted to try was a conditional DCGAN. We wanted to see how it would perform on this task given its success as a generative model for images. The cGAN is composed of an encoder and a generator and a discriminator.
The generator has three 2D convolutional layers with 3x3 kernels and stride 2, sandwiched between 2 fully connected layers. The first fully connected layer and each convolutional layer is followed by a ReLU layer. Channel depths after each convolutional layer are 16, 32, and 64. The generator takes in a gaussian noise vector of length 100, and the input 80x80 center crop, flattened into a vector. It outputs a vector of length 10*90*4 which we can map back to the frame of the ground truth image. This choice was made to reduce memory load.
The discriminator takes in the flattened 80x80 center crop given to the generator, as well as the flattened frame of either the ground truth or the generated frame. This flat vector is fed into a similar structure to the generator, with the same channel depths, kernel sizes, and strides, but with the last fully convolutional layer outputing a pair of scores for generated vs true, which we normalize using softmax.
The generator and discriminator are trained against each other with a zero sum GAN loss.

4 Results

(show results) (table of MSE and SSIM values) (image results) (?)

4.1 Linear Regression

(Merve)

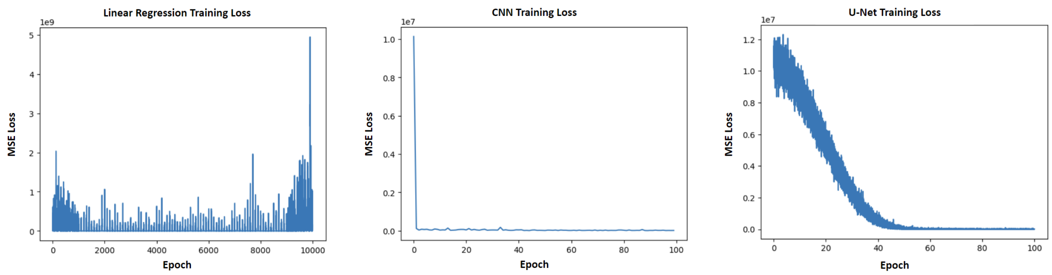

Figure 3: Loss Plots

Table 2: Model Performances

Model	MSE	SSIM
Linear Regression	5781.53	0.8381
CNN	11464.52	0.6955
U-Net	2216.99	0.9848
cGAN	0.3028	0.2174

4.2 Convolutional Neural Network

(Merve)

4.3 Pretrained U-Net

(Merve)

4.4 Conditional GAN

We trained the cGAN for 50 epochs with a learning rate of 0.02 and a standard Adam optimizer. For memory reasons, this was done as 5 checkpoints of 10 epochs. Gan loss plot is not meaningful, as both the generator and discriminator losses stay very near their starting values of 1.4 and 0.7 respectively for the entirety of training. As such, we do not show it to save space. Each checkpoint of 10 epochs took 3 hours to train, and despite this, the results are little better than simple gaussian noise [Fig. **??**] Note that the MSE seems small since the data is normalized to between 0 and 1 before being passed to the cGAN. The ssim gives a better approximation of the quality than the MSE.

5 Discussion

(compare methods and conclude which is best and why)

(Max?)

6 Conclusion

(Mike?)

7 Individual Contributions

- Merve Kilic:
  Implemented, trained, and ran some experiments for linear regression, CNN, U-Net.
- Maxime Ghesquiere:
  Write code for data cleaning and loading, cGAN, and loss curves. Train cGAN. Helped write code to display results.

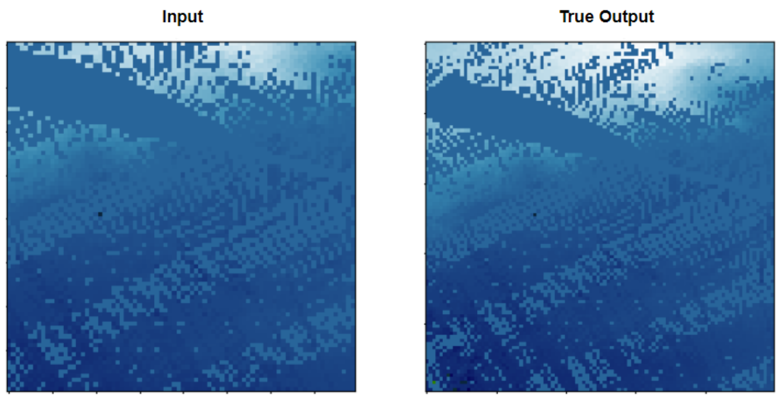

Figure 4: Testing input and true output

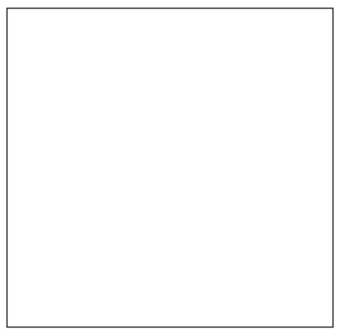

Figure 5: Sample figure caption.

- Mike Liu:
  data collection (manual), helped train CNN, helped write code to display results

References

[1] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. arXiv: `1505.04597 [cs.CV]`.