Deep-learning based ocean bottom seismic wavefield recovery

Ali Siahkoohi, Rajiv Kumar, and Felix J. Herrmann
Problem setup

Ocean bottom node (OBN) geometry:

- assume a desirable source sampling, via (simultaneous-source) randomized marine acquisition
- very sparse receivers scattered throughout the ocean bottom, **but** on a grid

**Objective:**

Reconstruct the information in the missing receivers
Why a neural net?

Most of previous methods rely on linear mathematical models:
► superposition of prototype waveforms from a fixed or learned dictionary or in terms of a matrix factorizations
► Particularly, matrix completion can be considered as a two-layer linear neural net

Using a nonlinear neural net, we find an implicit deep factorization
A supervised learning technique for wavefield reconstruction that does not need any external training data

i.e., training data is extracted from the acquired data
Seismic data in a 3D survey

Seismic data is 5D:

\[(t, \text{Src} \, x, \text{Src} \, y, \text{Rec} \, x, \text{Rec} \, y)\]

Taking Fourier transfer w.r.t. time:

\[(\omega, \text{Src} \, x, \text{Src} \, y, \text{Rec} \, x, \text{Rec} \, y)\]

Monochromatic seismic data is 4D:

\[(\text{Src} \, x, \text{Src} \, y, \text{Rec} \, x, \text{Rec} \, y)\]
Matricization of monochromatic seismic data

Our framework operators on monochromatic frequency slices

Two choices for matricization of monochromatic seismic data

- $\text{Rec } x, \text{Rec } y \times \text{Src } x, \text{Src } y$
- $\text{Rec } y, \text{Src } y \times \text{Rec } x, \text{Src } x$
Fully-sampled data
$(\text{Rec}_y, \text{Src}_y) \times (\text{Rec}_x, \text{Src}_x)$ domain
Observed data – Sampling rate 10% 
(Rec \(y, \) Src \(y\)) \(\times\) (Rec \(x,\) Src \(x\)) domain
Fully-sampled data
\((\text{Rec} \, x, \, \text{Rec} \, y) \times (\text{Src} \, x, \, \text{Src} \, y)\) domain
Observed data – Sampling rate 10%
\((\text{Rec } x, \text{ Rec } y) \times (\text{Src } x, \text{ Src } y)\) domain
Objective: Recovering missing receivers

Observed data - Sampling rate: 10% - Frequency: 10Hz

Reconstructed data - SNR: 23.46 dB
Proposed method

0. Pre-train a neural network  (more on this soon. For now, assume we have this)

1. Extract single-source frequency slices
   i.e., columns of \((\text{Rec } x, \text{ Rec } y) \times (\text{Src } x, \text{ Src } y)\)

2. Reconstruct the missing values by feeding the extracted slices to the pre-trained neural network
Proposed method:
Step 1: Extract and reshape
Proposed method:
Step 2: Reconstruction via the pre-trained neural net

$g_{\theta^*}$ is the pre-trained neural net.
Pre-training a neural net: Training data

Problem:

► we need training data pairs, i.e., subsampled and fully-sampled frequency slices

Solution:

► extract fully-sampled single-receiver frequency slices and subsample them with an arbitrary training mask

Underlying assumption:

► source-receiver reciprocity holds + dense source sampling
Steps to Extract training pairs

1. Extract and reshape single-receiver slices for existing receivers (fully sampled rows in \((\text{Rec } x, \text{ Rec } y) \times (\text{Src } x, \text{ Src } y)\) domain)
   - as many slices as recording receivers we have in the field
   - desired output of the network during training

2. Choose a training mask

3. Apply the training mask to artificially subsampled extracted single-receiver slices
   - input of the network during training
Training data: fully-sampled slices

Extract single-receiver slices for existing receivers, i.e., from acquired data
Training data: subsampled slices

What about the the input for supervised learning?

\[ G_\theta \]
Steps to Extract training pairs

1. Extract and reshape single-receiver slices for existing receivers (fully sampled rows in $(\text{Rec } x, \text{ Rec } y) \times (\text{Src } x, \text{ Src } y)$ domain)
   - as many slices as recording receivers we have in the field
   - desired output of the network during training

2. Choose a training mask

3. Apply the training mask to artificially subsampled extracted single-receiver slices
   - input of the network during training
Training data: subsampled slices

Arbitrarily subsample the extracted fully-sampled slice with a training mask.
Choosing training mask?

**Reminder:** The objective is to fill-in the columns

---

**Observed data - Sampling rate: 10% - Frequency: 10Hz**

- Source x, Source y

**Reshape**

- Partial measurements (10% sampling)

---

**Graph**

- Grid showing partial measurements with color coding for in-line and cross-line direction (m)
Training mask

We are free in choosing the training mask for artificial subsampling.

We choose a random training mask equal to the randomly missing receiver sampling mask.

► we know the missing pattern of receivers

Our experiments show that a random training mask is essential for successful wavefield recovery, even when receivers are on a periodic grid.
We use Generative Adversarial Networks (GANs) (Goodfellow et al., 2014)

GANs are based on an adversarial training procedure, i.e. involves two networks:
- Generator: is trained to reconstruct the artificially subsampled single-receiver slices
- Discriminator: is trained to distinguish between true single-receiver slices and reconstructed slices

We use a ResNet (He et al., 2016) based architecture for Generator and a fully-convolutional CNN with down-sampling for Discriminator.
Training framework: GANs

\[
\begin{align*}
\min_{\theta} \quad & \mathbb{E}_{x \sim p_X(x), y \sim p_Y(y)} \left[ (1 - D_\phi(G_\theta(x)))^2 + \lambda \| G_\theta(x) - y \|_1 \right], \\
\min_{\phi} \quad & \mathbb{E}_{x \sim p_X(x), y \sim p_Y(y)} \left[ (D_\phi(G_\theta(x)))^2 + (1 - D_\phi(y))^2 \right].
\end{align*}
\]

\{x, y\} \quad \text{Input/output pairs, drawn from the probability distributions } p_X(x) \text{ and } p_Y(y) \\
G_\theta(x) \quad \text{Generator} \\
D_\phi \quad \text{Discriminator}

\ell_1\text{-norm misfit term weighted by } \lambda \text{ ensures that each realization of } G_\theta(x) \text{ maps to a particular } y, \text{i.e., } x \mapsto y \text{ rather than solely fooling the discriminator.}
Testing Stage: reconstruction

Extract all the single-source slices (columns)
Testing Stage: reconstruction
Apply the trained neural network to all columns
Desirable source sampling, i.e., finely sampled sources

Source-receiver reciprocity holds under certain conditions

We hope Convolutional Neural Networks to perform well on testing data, i.e., reciprocal frequency slices

**does not need any external training data**
Dataset

Numerically simulated data on 3D BG Compass model
► $172 \times 172$ 2D periodic grid of sources
► $172 \times 172$ 2D periodic grid of receivers
► 25 m spatial sampling in both horizontal directions
► strong vertical and lateral variations

We processed the data for imaging by muting direct/turning waves
Numerical experiments

Applied to 3, 5, 10, and 15 Hz monochromatic data:

- missing 90% of receivers, randomly
- missing 90% of receivers, periodically

Training mask:

- experiments show that using a periodic training mask degrades the results
- for both cases (random and periodic), we train a single neural net using a random training mask
Fully-sampled data - 10 Hz
Observed data – Sampling rate 10%, randomly
Recovered data - 10 Hz - random case
\((\text{Rec } y, \text{ Src } y) \times (\text{Rec } x, \text{ Src } x)\) domain

Reconstructed data - SNR: 23.46 dB

Reconstruction error
Observed data – Sampling rate 10%, periodically
Recovered data - 10 Hz - periodic case
\((\text{Rec } y, \text{ Src } y) \times (\text{Rec } x, \text{ Src } x)\) domain

Reconstructed data - SNR: 20.83 \(\text{dB}\)

Reconstruction error
Fully-sampled data - 15 Hz
Observed data – Sampling rate 10%, randomly

Observed data - Sampling rate: 10% - Frequency: 15Hz

Receiver y, Source y

Receiver x, Source x

Observed data - Sampling rate: 10% - Frequency: 15Hz

Receiver x, Receiver y

Source x, Source y
Recovered data - 15 Hz - random case
(Rec y, Src y) × (Rec x, Src x) domain

Reconstructed data - SNR: 17.32 dB

Reconstruction error
Observed data – Sampling rate 10%, periodically

Observed data - Sampling rate: 10% - Frequency: 15Hz

Observed data - Sampling rate: 10% - Frequency: 15Hz
Recovered data - 15 Hz - periodic case 
\((\text{Rec } y, \text{ Src } y) \times (\text{Rec } x, \text{ Src } x)\) domain

Reconstructed data - SNR: 9.12 dB

Reconstruction error
Reconstruction quality

<table>
<thead>
<tr>
<th>Sampling mask</th>
<th>Frequency</th>
<th>Average recovery SNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>random</td>
<td>3 Hz</td>
<td>32.66 dB</td>
</tr>
<tr>
<td>random</td>
<td>5 Hz</td>
<td>29.07 dB</td>
</tr>
<tr>
<td>random</td>
<td>10 Hz</td>
<td>23.46 dB</td>
</tr>
<tr>
<td>random</td>
<td>15 Hz</td>
<td>17.31 dB</td>
</tr>
<tr>
<td>periodic</td>
<td>3 Hz</td>
<td>32.17 dB</td>
</tr>
<tr>
<td>periodic</td>
<td>5 Hz</td>
<td>28.32 dB</td>
</tr>
<tr>
<td>periodic</td>
<td>10 Hz</td>
<td>20.82 dB</td>
</tr>
<tr>
<td>periodic</td>
<td>15 Hz</td>
<td>9.12 dB</td>
</tr>
</tbody>
</table>

Table 1: Average reconstruction SNR for 90% random/periodic missing receivers.
Proposed method

vs

matrix completion method
Conclusions

The method does not need any external training data, assuming:

- source-receiver reciprocity
- desirable source sampling

Experiments show that random training mask is beneficial for recovery:

- missing either randomly, or periodically

Future work: perform FWI with data obtained by reconstructing low-frequency spectrum of the observed data.