GEOPHYSICS®

Cross-streamer wavefield reconstruction through wavelet domain learning

Journal:	*Geophysics*
Manuscript ID	GEO-2019-0771.R1
Manuscript Type:	Technical Paper
Keywords:	reconstruction, interpolation, neural networks, signal processing, inversion
Area of Expertise:	Signal Processing, Seismic Inversion

ABSTRACT

Seismic exploration in complex geological settings and shallow geological targets has led to a demand for higher spatial and temporal resolution in the final migrated image. Both conventional marine seismic and wide azimuth data acquisition lack near offset coverage, which limits imaging in these settings. A new marine source over cable survey, with split-spread configuration, known as TopSeis, was introduced in 2017 in order to address the shallow-target problem. However, wavefield reconstruction in the near offsets is challenging in the shallow part of the seismic record due to the high temporal frequencies and coarse sampling that leads to severe spatial aliasing. We investigate deep learning as a tool for the reconstruction problem, beyond spatial aliasing. Our method is based on a convolutional neural network (CNN) approach trained in the wavelet domain in order to reconstruct the wavefield across the streamers. We demonstrate the performance of the proposed method on broadband synthetic data and TopSeis field data from the Barents Sea. From our synthetic example, we show that the CNN can be learned in the inline direction and applied in the crossline direction, and that the approach

preserves the characteristics of the geological model in the migrated section. In addition, we compare our method to an industry-standard Fourier-based method, where the CNN approach shows an improvement in the root mean square (RMS) error close to a factor of two. In our field data example, we show that the approach manages to reconstruct the wavefield across the streamers in the shot domain, and displaying promising characteristics of a reconstructed 3D wavefield.

INTRODUCTION

Seismic processing and imaging of the subsurface in the Barents Sea is challenging for several reasons. The large uplift and erosion of the area manifest itself in rocks with high seismic velocities at shallow depth. These large velocity contrasts set up complex multiples and allows only a narrow cone of the reflected energy from the seismic sources to penetrate and illuminate the subsurface targets (Lie et al., 2018). The conventional 3D seismic spread lack the near offsets, which are important for imaging shallow parts of the subsurface and of great benefit for multiple attenuation (Vinje et al., 2017). A conventional 3D acquisition does therefore not represent an optimal setup in these environments. In order to address the shallow-target seismic-imaging issue, CGG and Lundin Norway proposed in 2017 a tailored acquisition solution, known as TopSeis (Vinje et al., 2017), yielding improved recording of the near offsets. This acquisition solution utilizes a split-spread, source-over-cable configuration, reduced streamer separation, wider source separation and the deployment of two or more sources. Still, near offsets are sparse, and suffer from spatial aliasing due to high temporal frequencies

and coarse sampling across the streamers. The interpolation and reconstruction problem of aliased data is considered an under-determined system, as there are infinite solutions to which the aliased seismic data could fit the dense. Hence, a priori information about the wavefield, such as seismic velocities, or assumptions concerning local linearity, sparsity or matrix rank are needed in order to reconstruct the wavefield. However, in the presence of severe aliasing these assumptions may not hold. In this paper, we investigate the potential use of deep learning as a tool to reconstruct the wavefield across the streamers in the shallow part of the seismic record where temporal frequencies are high and spatial sampling is coarse.

Several seismic interpolation methods exist. Wave-equation based methods (Fomel, 2003) may deal with irregular and regular sampling in presence of aliasing, but are computationally heavy and need a velocity model as a precondition. Some techniques assume local linearity and interpolate data in the frequency-space domain (Spitz, 1991; Porsani, 1999; Crawley, 2000; Naghizadeh and Sacchi, 2008). Some methods may reconstruct in the presence of mild degree of spatial aliasing under the assumption that

the interpolation problem becomes a matrix completion (rank-reduction) problem (Trickett et al., 2010), minimum weighted norm inversion of the subsampling operator (Liu and Sacchi, 2004) or least squares fitting of sinusoids (Ghaderpour et al., 2018; Ghaderpour, 2019). In case of high temporal frequencies and coarse sampling in the near offsets, where seismic events display conflicting dips and highly curved events, the linear assumption breaks down and these methods are therefore not optimal. Some methods deals with the interpolation problem by means of predefined sparse transforms, such as Fourier (Xu et al., 2005, 2010; Zwartjes and Sacchi, 2006; Schonewille et al., 2009; Naghizadeh and Sacchi, 2010a), Radon (Ibrahim et al., 2015), Curvelet (Naghizadeh and Sacchi, 2010b), Seislet (Gan et al., 2015) and Focal (Kutscha et al., 2010). The interpolation problem is then solved in combination with sparse optimization algorithms such as matching pursuit (Mallat and Zhang, 1993), basis pursuit (Boyd and Vandenberghe, 2004) or projection on to convex sets (Abma and Kabir, 2006). However, it is challenging to find a single transform that sparsifies all events such as diffractions and reflections in both shallow and deep, and some events are not optimally sparse (Turquais et al., 2018). Rather than relying on a predefined sparse transforms other

sparse approximation methods such as dictionary learning, which assume that the seismic data is a sparse linear combination of atoms defined from an over complete dictionary (Turquais et al., 2017, 2018; Zhu et al., 2017). Alternatively, complementary information from multicomponent data in combination with sparse optimization (Robertsson et al., 2008; Özdemir et al., 2010; Vassallo et al., 2010), crossline reconstruction beyond aliasing is possible. An alternative path to the seismic interpolation problem are the use of data driven approaches such as machine learning (Jia and Ma, 2017) and deep learning (Wang et al., 2018; Mandelli et al., 2019), which has drawn much attention recently. Deep learning based on CNN can be view as special case of the traditional sparse approximation based methods, but instead of optimizing each component separately, all components are jointly optimized (Dong et al., 2015), which allows for a fast and efficient data driven approach to the reconstruction problem.

Our approach to the seismic wavefield reconstruction problem may be seen as an analogy to inverse problems in image scaling where low-resolution digital images are transformed into their corresponding high-resolution counterparts. These set of

techniques are commonly referred to as single-image super-resolution (SISR), or image restoration, and dates back to the mid-eighties and work of Tsai and Huang (1984). Popular SR imaging methods can be categorized into four main types (Yang et al., 2014): prediction models (Irani and Peleg, 1991), edge based methods (Sun et al., 2010), statistical methods (He and Siu, 2011; Efrat et al., 2013) and example/patch/learning based methods (Chang et al., 2004). The deep learning-based methods have in recent years shown increased popularity in image reconstruction problems because of their improved efficiency and state-of-the-art performance compared to aforementioned methods (Yang et al., 2019). Deep learning approaches are specifically designed to automatically learn an end-to-end mapping between the low and high-resolution examples (Dong et al., 2015), where the non-linear representations are learned from large image databases.

In learning-based SISR, the low-resolution examples are commonly modelled from their corresponding high-resolution examples through blurring and downsampling followed by upsampling to the same size using conventional interpolation method, such

as bicubic interpolation (Dong et al., 2015; Guo et al., 2017). The SISR model is then applied to remove the blurring effect. To reduce computational complexity and to accelerate learning and application, the sub-optimal interpolation prior to learning can be skipped, and the upsampling operation is moved to within or toward the end of the network (Dong et al., 2016; Shi et al., 2016). In addition, methods such as bicubic interpolation do not bring additional information to the reconstruction problem (Shi et al., 2016). However, if the image contains complex structures and patterns, state-of-the-art CNNs applied directly in the image-domain are still inferior to other traditional methods (Bae et al., 2017). By transforming the low and high-resolution examples into a feature space prior to learning, such as wavelet-domain, increases the performance of the learning algorithm due to the simplification of the image structure (Bae et al., 2017; Guo et al., 2017; Liu et al., 2018).

The proposed wavefield reconstruction approach, which is inspired from two deep learning CNN-based techniques (Shi et al., 2016; Guo et al., 2017), considers the low (subsampled) and high-resolution (target) seismic gathers as training examples, and

transform them into the wavelet domain where we learn the upsampling function. The key advantages of this method are: (1) reduced computational complexity due to smaller size of the input in combination with time-space compression from the wavelet transform, (2) the wavelet domain simplifies the structure of the seismic gather, which could potentially increase performance in presence of complex structural patterns and (3) our approach is not dependent on any prior interpolation method to learn from, it is only limited to the available bandwidth in the domain from which we train the upsampling function. In addition, we hypothesize that the nonlinear relationships between the subsampled wavefields and their corresponding target wavefields can be learned along one direction –where the spacing between sensors are adequately dense– and then use the trained function to reconstruct the wavefield in a coarser direction (Greiner et al., 2019). In the synthetic data example, we use offset gathers in the inline direction as the dense cases, and attempt to reconstruct the much coarser crossline direction. In the field data example, we use marine shot gathers with split-spread geometry, illustrated in Figure 1, in the inline direction to reconstruct across the streamers where sampling is much coarser.

This paper is organized as follows: first, we introduce learning based SISR and discusses how SISR can be employed within seismic wavefield reconstruction. Here we also present the proposed methodology and approach. In the 2nd section, we demonstrate the performance of the method on synthetic broadband data and compare the approach to an industry-standard method. In the 3rd section, we apply the approach to field data from the Barents Sea by reconstructing the 3D wavefield in the shot domain. Finally, a discussion and a set of conclusions are given.

LEARNING-BASED WAVEFIELD RECONSTRUCTION

Learning-based super-resolution

The resolution limitation in seismic data is affected by the available bandwidth, the geometry and density of the sampling array. If a subset of the data is not sampled according to the Nyquist sampling criterion, aliasing will take place, as in the case for the undersampled crossline data. To model the crossline resolution problem, consider the underdetermined system of linear equations

$$\boldsymbol{DY} = \boldsymbol{X}, \tag{1}$$

where $\boldsymbol{Y} \in \mathbb{R}^{M \times N}$ denotes the target, represented by a seismic gather of size $M \times N$, where M is the number of time samples and N is the number of traces. The subsampled counterpart $\boldsymbol{X} \in \mathbb{R}^{M \times \frac{N}{r}}$ is decimated by the known operator $\boldsymbol{D}$:$\mathbb{R}^{M \times N} \rightarrow \mathbb{R}^{M \times \frac{N}{r}}$, where r is the subsampling factor. To illustrate the action of the operator $\boldsymbol{D}$, let $\boldsymbol{Y} \in \mathbb{R}^{3 \times 4}$ represent a two-dimensional seismic signal with four traces $[y_{0i}, y_{1i}, y_{2i}, y_{3i}]$ and three time samples

$i = 0,1,2$ downsampled by a ratio of $r = 2$. The decimation operation in equation (1), with $\boldsymbol{Y}$ represented in vector form, becomes

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} y_{00} \\ y_{01} \\ y_{02} \\ y_{10} \\ y_{11} \\ y_{12} \\ y_{20} \\ y_{21} \\ y_{22} \\ y_{30} \\ y_{31} \\ y_{32} \end{pmatrix} = \begin{pmatrix} x_{00} \\ x_{01} \\ x_{02} \\ x_{20} \\ x_{21} \\ x_{22} \end{pmatrix}. \quad (2)$$

It is obvious in equation (2) that the operator $\boldsymbol{D}$ is singular and has no inverse. Since seismic signals are highly structured and smoothly varying, we might still be able to estimate an inverse valid for this subspace. We try to establish this relation between the subsampled and the target seismic gathers, by the functional relationship

$$\boldsymbol{Y} = f(\boldsymbol{X}) + \varepsilon, \quad (3)$$

where ε represents the noise. In our case, we seek an approximate function $\hat{f}(\boldsymbol{X})$:$\boldsymbol{X} \in \mathbb{R}^{M \times \frac{N}{r}} \rightarrow \hat{\boldsymbol{Y}} \in \mathbb{R}^{M \times N}$ where $\hat{\boldsymbol{Y}}$ is the predicted seismic output from the learned function $\hat{f}(\cdot)$.

Figure 2 shows an example of a 2D shot gather in the inline direction, which is extracted from a 3D marine split-spread shot gather and cropped in time and space.

In case of shot-domain reconstruction across the cables, the 2D split-spread gathers along with the subsampled counterpart (Figure 2), will then define the training data $\{(\boldsymbol{Y}^{[i]},\boldsymbol{X}^{[i]})\}_{i=1}^{K}$ for K number of examples in the training data, from which we propose to learn the function $f(\cdot)$. Our assumption is that the decimated wavefield in the inline direction is representative for the wavefield in the crossline direction. We expect this to be a reasonable assumption, especially considering split-spread geometry (Figure 1), which yield near offset data giving similar inline-crossline characteristics, as seen in Figure 2b and 2d.

Employing wavelet-domain learning

Following Bae et al. (2017), Guo et al. (2017) and Liu et al. (2018), we propose to learn the function $f(\cdot)$ in the wavelet domain. In 2D discrete wavelet transform (DWT), digital filters are introduced that are characterized by their scaling function φ and wavelet

basis ψ. These are used to convolve a 2D signal $\boldsymbol{Y}$ into sub-band information represented by wavelet coefficients (Mallat, 1989). The simplest filters are the Haar filters, which could be represented by four 2D convolution kernels as

$$g_A = \frac{1}{2}\begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix}, g_H = \frac{1}{2}\begin{bmatrix}1 & 1\\ -1 & -1\end{bmatrix}, g_V = \frac{1}{2}\begin{bmatrix}1 & -1\\ 1 & -1\end{bmatrix}, g_D = \frac{1}{2}\begin{bmatrix}-1 & 1\\ 1 & -1\end{bmatrix}. \quad (4)$$

In a wavelet decomposition of level 1, the target 2D signal $\boldsymbol{Y} \in \mathbb{R}^{M \times N}$ is convolved with g_A, g_H, g_V, g_D yielding the four-band representations $\boldsymbol{d}^A, \boldsymbol{d}^H, \boldsymbol{d}^V, \boldsymbol{d}^D \in \mathbb{R}^{m \times n}$ respectively, where $m = M/2$ and $n = N/2$. The four-band representation is schematically illustrated in Figure 3**a** with their respective convolutional kernels. In equation (4) we see that the filter g_A acts as a low-pass filter on $\boldsymbol{Y}$, and is therefore commonly referred to as the approximation, while $\boldsymbol{d}^H$, $\boldsymbol{d}^V$ and $\boldsymbol{d}^D$ represent the horizontal, vertical and diagonal detail respectively. Figure 3**b** shows an example of a four-band split of a seismic source gather. We will use the following compact notation for forward wavelet transform and inverse wavelet transform

$$\tilde{\boldsymbol{Y}} = \boldsymbol{\psi}_{\mathrm{H}}\boldsymbol{Y} \text{ and } \tilde{\boldsymbol{X}} = \boldsymbol{\psi}_{\mathrm{H}}\boldsymbol{X}, \quad (5)$$

$$\boldsymbol{Y} = \boldsymbol{\psi}_{\mathrm{H}}^{*}\tilde{\boldsymbol{Y}} \text{ and } \boldsymbol{X} = \boldsymbol{\psi}_{\mathrm{H}}^{*}\tilde{\boldsymbol{X}}, \tag{6}$$

where subscript H denotes the Haar filter, the superscript * denotes the adjoint, and $\tilde{\boldsymbol{Y}} \in \mathbb{R}^{m \times n \times 4}$ and $\tilde{\boldsymbol{X}} \in \mathbb{R}^{m \times \frac{n}{r} \times 4}$ denote respectively the decomposition of the subsampled gather. In this case, $\tilde{\boldsymbol{X}}$ denotes the input to the CNN and $\tilde{\boldsymbol{Y}}$ denotes the target we want to predict from the learned function $\hat{\tilde{f}}(\cdot)$. Wavelet decomposition and reconstruction was performed using PyWavelet (Lee et al., 2019).

Convolutional neural network (CNN) architecture

Convolutional neural networks are a special case of artificial neural network architectures that incorporates knowledge about the invariance of the object shapes by using local connection patterns and by imposing constraints on the learnable parameters (LeCun et al., 1998). A CNN consists of L layers of linear convolutions and non-linear transforms. The computational transition from an arbitrary layer $l \in \{0,1,2,...,L\}$ to layer $l+1$ within the network consists of computing a set of convolution operations on the values $\boldsymbol{a}^{l}$ to give the linear output $\boldsymbol{z}^{l+1}$, where $\boldsymbol{a}^{l}$ and $\boldsymbol{z}^{l+1}$ can be considered as 3D

objects consisting of 2D features. The non-linear output is then given by non-linear transforms on $\boldsymbol{z}^{l+1}$ to give the values $\boldsymbol{a}^{l+1}$. A simple illustration of the convolution operation is given in Figure 4 where the input has two features. We will now discuss our network architecture, which is schematically represented in Figure 5. There are four types of transforms in our design:

1. Standard convolutional layers. These layers do not alter the dimension of the 2D features due to zero-padding, but might change the number of features.
2. Periodic upsampling to increase the 2D features size. This resampling operator upsamples the 2D feature space and reduces the number of features.
3. Non-linear transforms, i.e. activation functions.
4. Residual layers with single skip connections (He et al., 2016). Does not alter the dimension of the 2D features, but need same number of features for the input and residuals.

Let m denote the number of time samples, n the number of traces in the target, r the decimation factor, and c_l the number of features in layer l. The CNN is characterized

by a set of filters $\{\boldsymbol{w}_k^{l+1} \in \mathbb{R}^{\mathrm{f}\times\mathrm{f}\times c_l}\}_{k=1}^{c_{l+1}}$ and biases $\{b_k^{l+1} \in \mathbb{R}\}_{k=1}^{c_{l+1}}$ for each transition from a layer to the next. The dimension of the filters $\mathrm{f}\times\mathrm{f}$ denotes the size of the 2D convolution kernels and k is the feature number in the output, i.e. layer $l+1$. The linear output from the convolution operation of feature k can be written as

$$\boldsymbol{z}_k^{l+1} = \boldsymbol{w}_k^{l+1} * \boldsymbol{a}^l + b_k^{l+1}, \tag{7}$$

where $*$ denotes the convolution operation

$$\left(\boldsymbol{w}_k^{l+1} * \boldsymbol{a}^l\right)_{x,y} = \sum_{i=1}^{c_l} \sum_{u=-\mathrm{f}/2}^{\mathrm{f}/2} \sum_{v=-\mathrm{f}/2}^{\mathrm{f}/2} w_{u,v,i,k}^{l+1} a_{x-u,y-v,i}^{l}, \tag{8}$$

where the pair of indexes (x,y) defines each pixel position and the pair (u,v) define the kernel position. We can think of the convolution equation (8) as a special case of convolution, where the k^{th} output is computed by a sum of 2D convolutions, which implies employing a convolution kernel on each input feature and superimposed with the bias in each pixel position. Thus for a given layer there are as many biases as there are features in the next layer, i.e. c_{l+1}. The number of weights are dependent on the kernel size and the number of input and output features. To simplify the notations, let $\boldsymbol{W}^{l+1} \in$

$\mathbb{R}^{\mathrm{f}\times\mathrm{f}\times c_l\times c_{l+1}}$ and $\boldsymbol{b}^{l+1}\in\mathbb{R}^{c_{l+1}}$ denote all the weights and biases in layer $l+1$ respectively. As seen in Figure 5, the first layer weights have the dimensionality $7\times7\times4\times112$, where 7×7 denotes the kernel size, 4 denotes the number of input features, i.e. $\boldsymbol{d}^A,\boldsymbol{d}^H,\boldsymbol{d}^V,\boldsymbol{d}^D$, and 112 are the number of output features.

In order to introduce non-linearity, $\boldsymbol{z}^{l+1}=\left[\boldsymbol{z}_1^{l+1},...,\boldsymbol{z}_{c_{l+1}}^{l+1}\right]$ is passed through an activation function

$$\boldsymbol{a}^{l+1}=\phi(\boldsymbol{z}^{l+1}). \tag{9}$$

A popular approach in deep networks is to introduce layers with residual mapping, where activations from a previous layer is skipped over one layer and added to the next (single-skip). In case of residual mapping, a non-linearity in equation (9) is replaced by:

$$\boldsymbol{a}^{l+1}=\phi(\boldsymbol{z}^{l+1}+\boldsymbol{a}^{l-1}). \tag{10}$$

The main advantage of introducing residual layers is that it allows for training deeper networks, because optimizing the residual is easier than optimizing the target by a plain stack of non-linear layers (He et al., 2016). In our case, the non-linear function $\phi(\cdot)$ is

given by the Leaky Rectified Linear Unit (Maas et al., 2013) or Leaky ReLU, which is defined as

$$\phi(\boldsymbol{z}) = \max(\boldsymbol{z}, \alpha\boldsymbol{z}), \tag{11}$$

where $0 < \alpha < 1$ is the slope for negative arguments. In order to go from the subsampled space to the original 2D space we use a periodic upsampling operation within the network. We will refer to this layer as the spatial periodic resampling (SPR) layer, which in our case can be mathematically defined as

$$SPR(\boldsymbol{a})_{x,y,k} = \boldsymbol{a}_{\lfloor \frac{x}{r} \rfloor, y, c \cdot \mathrm{mod}(x,r) + k}, \tag{12}$$

where x, y and k are the pixel coordinates starting at $x,y,k = 0$, $\lfloor x/r \rfloor$ is integer division and $\mathrm{mod}(x,r)$ is the remainder of this division. From equation (12) we see that $SPR(\cdot)$: $\mathbb{R}^{m \times \frac{n}{r} \times cr} \rightarrow \mathbb{R}^{m \times n \times c}$ is an operator that simply rearranges the columns of $\boldsymbol{a}$ to upsample the 2D features by reducing the number of features, which is illustrated in Figure 6 resampling from $m \times \frac{6}{2} \times 4$ to $m \times 6 \times 2$.

Optimization problem

In deep learning, the optimization problem is typically solved by minimizing and/or maximizing an objective function using first-order gradient-based optimization algorithms, i.e. gradient descent/ascent, in combination with a learning-based procedure known as backpropagation (Rumelhart et al., 1988). Learning the weights $\boldsymbol{W}$ and biases $\boldsymbol{b}$ is then done by minimizing the following objective function:

$$\min_{\boldsymbol{W},\boldsymbol{b}} \{L(\tilde{\boldsymbol{Y}},\tilde{\boldsymbol{X}};\boldsymbol{W},\boldsymbol{b}) + \lambda R(\boldsymbol{W})\}, \tag{13}$$

where $L(\tilde{\boldsymbol{Y}},\tilde{\boldsymbol{X}};\boldsymbol{W},\boldsymbol{b})$ is the loss term, $R(\boldsymbol{W})$ is the regularization term and λ is the regularization parameter defining the tradeoff between the loss and the regularization. In order to minimize equation (13) by a gradient-based method we need to compute the partial derivatives w.r.t each weight and bias within the network. The optimization problem was solved by stochastic gradient descent using subsamples (mini-batches) using a version of the Adam optimizer defined in (Loshchilov and Hutter, 2018) as

$$\boldsymbol{W}^l_{[t]} = \boldsymbol{W}^l_{[t-1]} - \eta\left(\nabla_{\boldsymbol{W}^l} L(\tilde{\boldsymbol{Y}},\tilde{\boldsymbol{X}};\boldsymbol{W},\boldsymbol{b})_{[t]} + \lambda \nabla_{\boldsymbol{W}^l} R(\boldsymbol{W})_{[t-1]}\right), \tag{14}$$

where η is the fixed learning rate, $\nabla_{\boldsymbol{W}^l} R(\boldsymbol{W})$ is the partial derivatives of the regularization term and $\nabla_{\boldsymbol{W}^l} L(\tilde{\boldsymbol{Y}},\tilde{\boldsymbol{X}};\boldsymbol{W},\boldsymbol{b})$ is the partial derivatives of the loss using the Adam optimizer rule (Kingma and Ba, 2014). In our case, the loss in equation (13) is given by the L_1 norm $L(\tilde{\boldsymbol{Y}},\tilde{\boldsymbol{X}};\boldsymbol{W},\boldsymbol{b}) = \left\|\tilde{\boldsymbol{Y}} - \hat{\tilde{\boldsymbol{f}}}(\tilde{\boldsymbol{X}};\boldsymbol{W},\boldsymbol{b})\right\|_1^1$ and the regularization on the weights is given by the L_2 norm penalty $R(\boldsymbol{W}) = \frac{1}{2}\|\boldsymbol{W}\|_2^2$. The biases are updated in the same manner excluding the regularization term. Neural network implementation and training was performed in python using the Numpy and Tensorflow libraries (Abadi et al., 2016).

Network design, training and regularization

The optimization problem in deep learning is challenging due to being non-convex, overparameterized and unstable. Careful selection of hyperparameters, parameter initialization, and network design through trial and error is necessary in order to get the model to train and converge to a proper local minimum. Initialization of the weights were done using He-initialization (He et al., 2015), which is particularly designed for dealing with rectifier nonlinearities, and the biases were initialized to zero values. The hyper-parameters of the Adam optimizer were set to default, as we observed no improvements

by using other values. The network depth and width (number of layers and number of filters in each layer) were found by trial and error, along with the rectifier nonlinearities slope α and learning rate η, which were set to 0.1 and 0.001 respectively.

The computational model and network design used in our study –summarized and schematically illustrated in Figure 5– consists of three single-skip connection residual layers in subsampled 2D space, an SPR operation after layer 8 for upsampling followed by two non-linear layers and a standard linear output. The two first convolution layers, in addition to the two convolution layers after upsampling, consists of 7×7 followed by 5×5 kernels. The other convolution layers have 3×3 kernels. Similar to Dong et al., (2015), we found that by using larger kernel size in the start of the network improved the reconstruction and structural definition of the prediction. However, since larger kernels increase computational complexity, the choice of kernel size should be a tradeoff between performance and speed.

Prior to training, the training data are split randomly into three datasets, which we will refer to as the training, validation and test set. Each example in all sets were scaled to

have a maximum absolute value of 1. The training set is used to fit the model to the data, the validation set is used for model selection, i.e. selection of the regularization parameter λ, and the test set is used for model assessment for final evaluation.

For model selection, we used a cross-validation (CV) approach in combination with an early stopping algorithm. The goal of regularization and early stopping is to avoid overfitting in an overparameterized model. The early stopping approach avoids overfitting by storing the parameters that shows the best performance in one training phase, and stopping after a predefined number of iterations if the validation loss has not decreased. Regularization constrains the size of the weight parameters during training, which depends on the value of regularization parameter λ. CV is performed by training with different λ values, i.e. different models, and selecting the model that shows best performance on the validation set. We used a selection of $\lambda \in \{10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 0\}$, where $\lambda = 0$ and $\lambda = 10^{-3}$ denotes highest- and lowest model complexity respectively. We found $\lambda = 10^{-5}$ to be a good choice in both the synthetic example and field example.

An example from the loss during training from the field data example is shown in Figure A-1 in Appendix A.

SYNTHETIC DATA EXAMPLE

In order to evaluate the performance of our wavelet CNN and inline- learning to crossline-reconstruction approach, we have designed a synthetic 3D model and applied a diffraction modelling method similar to what is described in Jaramillo and Bleistein (1999) to create synthetic broadband data. For comparison, we use a 3D Fourier-based interpolation method (Xu et al., 2005), which is similar to currently applied approaches in the industry. We will refer to this method as 3D Fourier in the following.

As can be seen from Figure 7, the model consists of a rugose water bottom from bathymetry in the Barents Sea and a 2D grid of equally spaced diffraction lines at a depth of $518\ \mathrm{m}$. The water bottom and diffraction lines in this model will give a complex data set with strong aliasing and conflicting dips which is a challenge to any interpolation algorithm.

In Figure 8a, we show the survey setup, where we model 64 constant-offset, zero-azimuth data cubes on a $6.25 \times 6.25\ \mathrm{m}$ grid with offset spacing of $12.5\ \mathrm{m}$ and a time

sampling of $2\ \mathrm{ms}$. The source signal is zero phase with a flat frequency spectrum from 4 to 175 Hz. We assume no external noise in this experiment, constant velocity of 1480 m/s and no anelastic attenuation.

In the following, we will refer to the densely sampled $6.25 \times 6.25\ \mathrm{m}$ cubes as the ground truth. We then decimate the ground truth to create a new subsampled data set, where we have removed three of four traces to get a sampling of $6.25 \times 25\ \mathrm{m}$ as shown in Figure 8**b**. An example of the ground truth and the subsampled counterpart used for training in the inline direction, zoomed at the water bottom and diffraction lines are displayed in Figure 9. The subsampled data defines the training data $D = \{(\boldsymbol{Y}^{[i]}, \boldsymbol{X}^{[i]})\}_{i=1}^{K}$, where $\boldsymbol{Y}$ is an offset class along one inline and $\boldsymbol{X}$ is the subsampled counterpart, consists of 162 inlines, which gives $K = 10368$ examples for the 64 offset classes. Training data size were doubled by augmentation, by flipping the data to give negative offsets. A subset of $K = 15900$ examples were then used as training data, which were further split into training set and validation set of 14900 and 1000 respectively. The crossline section will

then represent the test set for final assessment. Using 400 time samples, the size of each training example is 400×640 and 400×160 for the target and subsampled data respectively. The wavelet CNN is then trained along the inline direction where the sampling is dense and applied to the coarsely sampled crossline direction. The training phase run for 58700 iterations (19 epochs) before early stopping initiated, which took approximately 10 hours on a modern GPU (NVIDIA Tesla V100 SXM2 32GB). When the training phase is complete the run-time for each crossline section (600×162) takes approximately 0.2 s.

From the ground truth in Figure Figure 9**a** and 10**a** we observe that the rugose water bottom creates a complex reflected wavefield with myriads of diffraction-like events causing strong interference. Deeper down, at around 0.7 s in Figure 11**a**, we see the regular hyperbolic events from the diffraction lines parallel to the inlines, and the linear events from the diffraction lines parallel to the crosslines. The decimated data shown in Figure 10**b** and 11**b** is strongly aliased and the 3D Fourier interpolation shown in Figure 10**c** and 11**c** struggles to resolve the conflicting dips and the aliasing. The wavelet CNN

in Figure 10**e** and 11**e** does a much better job in reconstructing conflicting dips and aliased events, with some minor 'striping' artifacts. These 'striping' artifacts represent input traces where wavelet CNN have not managed to learn a suitable pattern for reconstruction. In case of reconstruction of the deeper part, the diffraction lines are equal in both inline and crossline direction, the model is therefore ideal for our assumptions of interchanging inline and crossline and gives an optimistic impression in contrast to the top inference, which is realistic.

To compare the two different methods ability to reproduce and preserve the geological features and patterns, we migrate the ground truth and the results from the 3D Fourier interpolation and wavelet CNN reconstruction, using Kirchhoff migration, which are displayed in Figure 12. We see that compared to the 3D Fourier interpolation, the wavelet CNN reconstruction shows less footprints, clearer structural definition and less migration noise. For a quantitate measurement and comparison we compute the RMS error and peak signal to noise ratio (PSNR) of the entire interval of the pre and post migrated sections, which are listed in Table 1. The PSNR of the data $\boldsymbol{d}$ is computed as

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{\max\left(\boldsymbol{d}_{\mathrm{ref}}^2\right)}{\frac{1}{M}\|\boldsymbol{d}_{\mathrm{ref}} - \boldsymbol{d}\|_2^2}\right). \quad (15)$$

where $\boldsymbol{d}_{\mathrm{ref}}$ is the reference data and M is the number of samples. Here we see that the wavelet CNN RMS error shows an improvement close to a factor of two compared to the 3D Fourier RMS error, which implies a considerable uplift for the CNN approach.

Table 1. RMS error and PSNR from the pre and post migrated crossline section from the entire interval.

	3D Fourier		Wavelet CNN	
	RMS	PSNR	RMS	PSNR
Pre migration	0.0086	41.3434	0.0047	46.4906
Post migration	0.0124	38.1423	0.0068	43.3990

FIELD DATA EXAMPLE

We tested the proposed method on TopSeis (Vinje et al., 2017) field data from the Barents Sea. The seismic data contained 901 3D shot gathers displaying complex wavefield setting in addition to a difficult noise setting. Each shot was recorded by 14 streamers separated by approximately 50 m, employing a temporal sampling rate of 2 ms and receiver separation along the streamer of 12.5 m. Each 3D shot gather were split into 14 2D shot gathers, one for each cable. The 2D shot gathers were cropped to a size of 250×80 (number of temporal- and spatial samples) and decimated in order to adapt to the crossline sampling interval for input. The subsampled and target gathers define the training data $D = \{(\boldsymbol{Y}^{[i]}, \boldsymbol{X}^{[i]})\}_{i=1}^{K}$ for $K = 12614$ training examples. In addition, we used data augmentation in order to increase the size of the training set, first by horizontal translation and second by horizontal flipping, increasing the number of examples in the training set to $K_{\text{train}} = 72648$. An example of subsampled-vs-target gathers with corresponding frequency-wavenumber plot is displayed in Figure 13. In Figure 13**d**, we see that the subsampled example with a 50 m spatial sampling interval gives rise to

severe aliasing in the frequency-wavenumber domain compared to the target example in Figure 13**c**. The subsampled example in Figure 13**b** is displayed with nearest neighbor interpolation, only for visual comparison. We extracted one example from the test set to visually compare the reconstructed and target gather. The target, subsampled and reconstructed gathers along with the reconstruction error and the RMS-amplitude spectrum, are shown in Figure 14. The DWT of the target and predicted DWT from the test set in Figure 14 are displayed in Figure C-1 in Appendix C.

In this example, the model is capable to reconstruct a complex wavefield with difficult structural patterns. Overall, the reconstruction error increases with depth where the 2D structure becomes more complex. The model struggles with frequencies above 50 Hz, and at high wavenumbers, which we can observe in the corresponding frequency-wavenumber spectra in Figure 14. Similar to the synthetic example we observe 'striping' artifacts in this example, which are present in the deeper part of the seismic record.

The loss of frequencies above $50\ Hz$ implies a smoother reconstruction than the target. However, we would expect the model to struggle with the high-frequency part due

to both complex noise setting. In order to have a quality measure between the different data sets, we computed the RMS error and PSNR. The average RMS error and PSNR of the three data sets are listed in Table 2. The RMS error and PSNR of the example in Figure 14 is 0.0497 and 26.0673 respectively. The RMS error and PSNR for all the shots in the test set are displayed in a scatterplot in Figure B-1 in Appendix B. The RMS error from the test set reconstruction is close to the RMS error in the validation set, which implies a good generalization from the proposed method. The training set shows a lower RMS error, which we expect since the model is fitted on this data set.

Table 2. Average RMS error and PSNR comparison of the different sets.

Test set		Validation set		Training set	
RMS	PSNR	RMS	PSNR	RMS	PSNR
0.0448	27.1471	0.0449	27.1318	0.0442	27.2670

The method manages to unwrap the signal at high frequencies and wavenumbers for curved events, linear events and conflicting dips, in addition to preserving the characteristics of the wavefield in 3D. The characteristics of the 3D wavefield is more evident on time slices. We sorted the 3D shot gather presented in Figure 16 into time slices, and extracted four slices corresponding to $t \in \{460\,, 480\,, 500\,, 520\,\}\,\mathrm{ms}$. The timeslices –before and after reconstruction– are shown in in Figure 17, where we observe in Figure 17**e** to 17**h** a high resolution wavefield with clear wavefield patterns implying a promising 3D wavefield reconstruction from the proposed method. Still, we observe the same 'striping' artifacts and loss of high frequencies as seen the inline case and the synthetic cases.

DISCUSSION

One advantage of the proposed method is that there is no need for manual labeling, pre-interpolation to learn from or any interpretation, leading to minimal human interference. Once the wavelet domain CNN is learned, reconstruction of the 3D wavefield is achieved automatically by applying to individual 2D crossline sections. In our approach, the introduction of wavelet transform adds additional complexity to the CNN approach as the wavelet basis of choice is to be considered an additional network hyperparameter. We experimented with different types of wavelets from the DWT filter bank, such as Haar, coiflets and symlets, where the Haar basis gave more stable results than the other wavelet basis. The stability of the Haar basis might relate to its short support, i.e. length of the wavelet, since the crossline section in the shot domain has only 14 traces for input. We could also consider using different wavelet basis in each direction, i.e. Haar in the space domain and another discrete –or continuous– wavelet in the temporal direction, which could potentially improve the method in presence of low signal to noise ratio.

The method is not dependent on any prior knowledge about the wavefield, such as velocities, and can therefore be trained on data comprised of multiple and ghost reflections. This implies a broader use of the method, such as a tool for wavefield reconstruction in an earlier stage of the processing flow, and therefore potentially used in demultiple and deghosting workflows and prior to migration. Early stage processing potential might be a topic for further investigation for application purposes. However, for the method to work optimally it require that the statistical properties of the wavefield in the inline direction are representative for the crossline direction.

As shown in the field data example in Figure 13 and 14, the method suffers from loss of energy at high frequencies and wavenumbers, which optimally would be restored by the model. From our tests, the introduction of wavelet domain learning contributed slightly to restoring more of the high frequencies. Other tests, by using different objective functions such as L_2 or L_1 loss in combination with L_1 or L_2 regularization, the L_1 loss gave the most significant improvement in reconstruction at high frequencies. Similar results were reported by Zhao et al. (2016), where they investigated the difference between L_2 and L_1 loss in image restoration problems. They argued that the L_2 loss gets more easily

stuck in a local minimum while the L_1 loss may be guided towards a better minimum, most likely due to the smoothness of the function and local convexity properties of the L_2 vs. L_1. If the problem is related to a difficult noise setting, a potential solution could be to introduce more data in order to reduce uncertainty and increase robustness in presence of noise. Alternatively, preprocessing of the data to remove some of the noise before training could also be a potential option.

Another potential challenge is in situations where streamer feathering and/or streamer fanning causes large deviations in streamer spacing. The CNN approach is dependent on the learned characteristics of the wavefield given by the relationship between target and subsampled counterpart. Augmenting the training data by subsampling with different trace spacing can potentially improve its robustness in these circumstances. Another potential challenge is the well-known problem in deep-learning for image super-resolution are reconstruction artifacts caused from instabilities due to carefully constructed noise, i.e. adversarial attack (Antun et al., 2019). As we have not

considered problems related to aforementioned challenges, this might be a potential topic for further research.

Concerning our hypothesis –using the densely sampled inline direction to learn non-linear representations for reconstruction of the wavefield in the crossline direction– we consider the seismic wavefield as highly structured and smooth 3D signals represented by local geometrically shaped patterns. Even though the wavefield represents high complexity in terms of aliasing, conflicting dips the seismic structure is determined by wave phenomena rather than geology. In this case, even though the inline and crossline differ in terms of geological features and patterns, we consider it reasonable that the wavefield patterns and features are locally similar in both directions. A challenge in field data is where the wavefield structures and patterns could be damaged by low S/N ratio and processing artifacts from various workflows down the processing pipeline. Differences in anisotropy, dispersion and other wave phenomena effects in addition to large difference in structural dip could also potentially play a role for the inline-to-crossline application. All these effects, which are not considered in our synthetic example, increase

the complexity and difficulty of learning a robust representation for wavefield reconstruction in different directions. In order to address the challenges related to inline-to-crossline application, a much more comprehensive synthetic example should be employed, by introducing realistic noise, more wave phenomena complexity and strong structural differences in inline compared to crossline, followed by a complete processing flow, in order to produce a total imaging impact of the approach. However, this type of study is beyond the scope of this paper, and is therefore planned for future research.

CONCLUSION

In this paper, we propose to use a deep CNN, trained in the wavelet domain, as a tool for seismic wavefield reconstruction beyond aliasing, where non-linear wavefield representations are learned on densely sampled seismic gathers –in the inline direction– in order to learn an upsampling function, which is applied on the undersampled wavefield in the crossline direction. We tested the wavelet domain based CNN method on synthetic data and field data where it manages to unwrap high frequencies at high wavenumbers.

In the synthetic case we compared the wavelet CNN to a Fourier-based industry standard method, where the wavelet CNN approach gave an improvement in RMS error close to a factor of 2. The deep-learning approach in this paper has proven to be an effective tool for seismic wavefield reconstruction, and an approach that allows for a fast and efficient reconstruction of aliased seismic data. However, to improve robustness of the model, more data should be included by introducing more seismic gathers in combination with different augmentation approaches. In addition, solutions related to the loss of high frequencies and wavenumbers is yet to be solved, and should be considered in future research.

APPENDIX A

TRAINING AND VALIDATION LOSS DURING TRAINING FOR FIELD DATA

An example from the loss during training from the training set batches (black curve) and validation set (red curve) is displayed in Figure A-1 with logarithmic scale on the x-axis. In this example, early stopping was set to train for 20000 iterations without improvement in the validation loss before initiating.

APPENDIX B

RMS AND PSNR PLOTS FOR FIELD DATA

The PSNR and RMS for all shots in the test set are displayed in Figure B-1, with the larger blue triangle indicating the test set shot shown in Figure 14.

APPENDIX C

WAVELET DOMAIN

The DWT of the target and predicted wavelet domain model from the test set shown in Figure 14 are displayed in Figure C-1.

REFERENCES

Abadi, M., A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, and M. Devin, 2016, Tensorflow: Large-scale machine learning on heterogeneous distributed systems: ArXiv Preprint ArXiv:1603.04467.

Abma, R., and N. Kabir, 2006, 3D interpolation of irregular data with a POCS algorithm: Geophysics, **71**, E91–E97.

Antun, V., F. Renna, C. Poon, B. Adcock, and A. C. Hansen, 2019, On instabilities of deep learning in image reconstruction-Does AI come at a cost? ArXiv Preprint ArXiv:1902.05300.

Bae, W., J. Yoo, and J. Chul Ye, 2017, Beyond deep residual learning for image restoration: Persistent homology-guided manifold simplification: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 145–153.

Boyd, S., and L. Vandenberghe, 2004, Convex Optimization: Cambridge university press.

Chang, H., D.-Y. Yeung, and Y. Xiong, 2004, Super-resolution through neighbor embedding: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004., **1**, I–I.

Crawley, S., 2000, Seismic Trace Interpolation with Nonstationary Prediction-Error Filters: Stanford University.

Dong, C., C. C. Loy, and X. Tang, 2016, Accelerating the super-resolution convolutional neural network: European Conference on Computer Vision, 391–407.

Dong, C., C. C. Loy, K. He, and X. Tang, 2015, Image super-resolution using deep convolutional networks: IEEE Transactions on Pattern Analysis and Machine Intelligence, **38**, 295–307.

Efrat, N., D. Glasner, A. Apartsin, B. Nadler, and A. Levin, 2013, Accurate blur models vs. image priors in single image super-resolution: Proceedings of the IEEE International Conference on Computer Vision, 2832–2839.

Fomel, S., 2003, Seismic reflection data interpolation with differential offset and shot continuation: Geophysics, **68**, 733–744.

Gan, S., S. Wang, Y. Chen, Y. Zhang, and Z. Jin, 2015, Dealiased seismic data interpolation using seislet transform with low-frequency constraint: IEEE Geoscience and Remote Sensing Letters, **12**, 2150–2154.

Ghaderpour, E., 2019, Multichannel antileakage least-squares spectral analysis for seismic data regularization beyond aliasing: Acta Geophysica, **67**, 1349–1363.

Ghaderpour, E., W. Liao, and M. P. Lamoureux, 2018, Antileakage least-squares spectral analysis for seismic data regularization and random noise attenuation: Geophysics, **83**, V157–V170.

Greiner, T. L., O. Kolbjørnsen, J. E. Lie, E. H. Nilsen, A. K. Evensen, and L. Gelius, 2019, Cross-streamer wavefield interpolation using deep convolutional neural network, *in* SEG Technical Program Expanded Abstracts 2019, Society of Exploration Geophysicists, 2207–2211.

Guo, T., H. Seyed Mousavi, T. Huu Vu, and V. Monga, 2017, Deep wavelet prediction for image super-resolution: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 104–113.

He, H., and W.-C. Siu, 2011, Single image super-resolution using Gaussian process regression: CVPR 2011, 449–456.

He, K., X. Zhang, S. Ren, and J. Sun, 2015, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification: Proceedings of the IEEE International Conference on Computer Vision, 1026–1034.

He, K., X. Zhang, S. Ren, and J. Sun, 2016, Deep residual learning for image recognition: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770–778.

Ibrahim, A., M. D. Sacchi, and P. Terenghi, 2015, Wavefield reconstruction using a Stolt-based asymptote and apex shifted hyperbolic Radon transform, *in* SEG Technical Program Expanded Abstracts 2015, Society of Exploration Geophysicists, 3836–3841.

Irani, M., and S. Peleg, 1991, Improving resolution by image registration: CVGIP: Graphical Models and Image Processing, **53**, 231–239.

Jaramillo, H. H., and N. Bleistein, 1999, The link of Kirchhoff migration and demigration

to Kirchhoff and Born modeling: Geophysics, **64**, 1793–1805.

Jia, Y., and J. Ma, 2017, What can machine learning do for seismic data processing? An interpolation application: Geophysics, **82**, V163–V177.

Kingma, D. P., and J. Ba, 2014, Adam: A method for stochastic optimization: ArXiv Preprint ArXiv:1412.6980.

Kutscha, H., D. J. Verschuur, and A. J. Berkhout, 2010, High resolution double focal transformation and its application to data reconstruction, *in* SEG Technical Program Expanded Abstracts 2010, Society of Exploration Geophysicists, 3589–3593.

LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner, 1998, Gradient-based learning applied to document recognition: Proceedings of the IEEE, **86**, 2278–2324.

Lee, G. R., R. Gommers, F. Waselewski, K. Wohlfahrt, and A. O'Leary, 2019, PyWavelets: A Python package for wavelet analysis: Journal of Open Source Software, **4**, 1237.

Lie, J. E., V. Danielsen, P. E. Dhelie, R. Sablon, R. Siliqi, C. Grubb, V. Vinje, C. I. Nilsen, and R. Soubaras, 2018, A Novel Source-Over-Cable Solution To Address

The Barents Sea Imaging Challenges: Marine Acquisition Workshop 2018.

Liu, B., and M. D. Sacchi, 2004, Minimum weighted norm interpolation of seismic records: Geophysics, **69**, 1560–1568.

Liu, P., H. Zhang, K. Zhang, L. Lin, and W. Zuo, 2018, Multi-level wavelet-CNN for image restoration: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 773–782.

Loshchilov, I., and F. Hutter, 2018, Decoupled weight decay regularization: .

Maas, A. L., A. Y. Hannun, and A. Y. Ng, 2013, Rectifier nonlinearities improve neural network acoustic models: Proc. Icml, **30**, 3.

Mallat, S. G., 1989, A theory for multiresolution signal decomposition: the wavelet representation: IEEE Transactions on Pattern Analysis & Machine Intelligence, 674–693.

Mallat, S. G., and Z. Zhang, 1993, Matching pursuits with time-frequency dictionaries: IEEE Transactions on Signal Processing, **41**, 3397–3415.

Mandelli, S., V. Lipari, P. Bestagini, and S. Tubaro, 2019, Interpolation and Denoising of

Seismic Data using Convolutional Neural Networks: ArXiv Preprint ArXiv:1901.07927.

Naghizadeh, M., and M. D. Sacchi, 2008, f-x adaptive seismic-trace interpolation: Geophysics, **74**, V9–V16.

Naghizadeh, M., and M. D. Sacchi, 2010a, On sampling functions and Fourier reconstruction methods: Geophysics, **75**, WB137–WB151.

Naghizadeh, M., and M. Sacchi, 2010b, Hierarchical scale curvelet interpolation of aliased seismic data, *in* SEG Technical Program Expanded Abstracts 2010, Society of Exploration Geophysicists, 3656–3661.

Özdemir, K., A. Özbek, D.-J. van Manen, and M. Vassallo, 2010, On data-independent multicomponent interpolators and the use of priors for optimal reconstruction and 3D up/down separation of pressure wavefields: Geophysics, **75**, WB39–WB51.

Porsani, M. J., 1999, Seismic trace interpolation using half-step prediction filters: Geophysics, **64**, 1461–1467.

Robertsson, J. O., I. Moore, M. Vassallo, K. Özdemir, D.-J. van Manen, and A. Özbek,

2008, On the use of multicomponent streamer recordings for reconstruction of pressure wavefields in the crossline direction: Geophysics, **73**, A45–A49.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams, 1988, Learning representations by back-propagating errors: Cognitive Modeling, **5**, 1.

Schonewille, M., A. Klaedtke, A. Vigner, J. Brittan, and T. Martin, 2009, Seismic data regularization with the anti-alias anti-leakage Fourier transform: First Break, **27**.

Shi, W., J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, 2016, Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1874–1883.

Spitz, S., 1991, Seismic trace interpolation in the FX domain: Geophysics, **56**, 785–794.

Sun, J., Z. Xu, and H.-Y. Shum, 2010, Gradient profile prior and its applications in image super-resolution and enhancement: IEEE Transactions on Image Processing, **20**, 1529–1542.

Trickett, S., L. Burroughs, A. Milton, L. Walton, and R. Dack, 2010, Rank-reduction-

based trace interpolation, *in* SEG Technical Program Expanded Abstracts 2010, Society of Exploration Geophysicists, 3829–3833.

Tsai, R. Y., Huang, T. S., 1984, Multiple frame image restoration and registration, *in* Advances in Computer Vision and Image Processing, , 317–339.

Turquais, P., E. Asgedom, and W. Söllner, 2017, Structured dictionary learning for interpolation of aliased seismic data, *in* SEG Technical Program Expanded Abstracts 2017, Society of Exploration Geophysicists, 4257–4261.

Turquais, P., E. G. Asgedom, W. Söllner, and L. Gelius, 2018, Parabolic dictionary learning for seismic wavefield reconstruction across the streamers: Geophysics, **83**, V263–V282.

Vassallo, M., A. Özbek, K. Özdemir, and K. Eggenberger, 2010, Crossline wavefield reconstruction from multicomponent streamer data: Part 1—Multichannel interpolation by matching pursuit (MIMAP) using pressure and its crossline gradient: Geophysics, **75**, WB53–WB67.

Vinje, V., J. E. Lie, V. Danielsen, P. E. Dhelle, R. Silliqi, C.-I. Nilsen, E. Hicks, and A.

Camerer, 2017, Shooting over the seismic spread: First Break, **35**, 97–104.

Wang, B., N. Zhang, W. Lu, and J. Wang, 2018, Deep-learning-based seismic data interpolation: A preliminary result: Geophysics, **84**, V11–V20.

Xu, S., Y. Zhang, and G. Lambaré, 2010, Antileakage Fourier transform for seismic data regularization in higher dimensions: Geophysics, **75**, WB113–WB120.

Xu, S., Y. Zhang, D. Pham, and G. Lambaré, 2005, Antileakage Fourier transform for seismic data regularization: Geophysics, **70**, V87–V95.

Yang, C.-Y., C. Ma, and M.-H. Yang, 2014, Single-image super-resolution: A benchmark: European Conference on Computer Vision, 372–386.

Yang, W., X. Zhang, Y. Tian, W. Wang, J.-H. Xue, and Q. Liao, 2019, Deep learning for single image super-resolution: A brief review: IEEE Transactions on Multimedia.

Zhao, H., O. Gallo, I. Frosio, and J. Kautz, 2016, Loss functions for image restoration with neural networks: IEEE Transactions on Computational Imaging, **3**, 47–57.

Zhu, L., E. Liu, and J. H. McClellan, 2017, Joint seismic data denoising and interpolation with double-sparsity dictionary learning: Journal of Geophysics and Engineering,

14, 802–810.

Zwartjes, P. M., and M. D. Sacchi, 2006, Fourier reconstruction of nonuniformly sampled, aliased seismic data: Geophysics, **72**, V21–V32.

LIST OF FIGURES

Figure 1. An illustration of a split-spread design employing both a streamer vessel source vessel with three sources.

Figure 2. An illustration of inline training data and the crossline reconstruction problem. The two panels to the left, **a**) and **b**), show a single training example in the shot domain where **a**) the target in the inline direction is subsampled to give **b**) the crossline representation. The inline targets and the subsampled counterparts in the training data are then used to learn the function $f(\cdot)$, in order to reconstruct **c**) the crossline target from **d**) the undersampled crosslines using the learned function $\hat{f}(\cdot)$.

Figure 3. A simplified illustration of the wavelet decomposition, where **a**) is a schematic representation of the four-band split with their respective convolution kernels ($\text{gray} = 1/2,\ \text{white} = -1/2$) and **b**) an example where a seismic source gather is decomposed using the Haar filter.

Figure 4. Simplified illustration of the convolution operation on an input with two features $\mathbf{a}_1$and $\mathbf{a_2}$ being convolved with a 2D kernel and superimposed with the bias parameter, to give the linear transform $\mathbf{z}_1$. A different set of 2D kernels and biases are then used to compute the linear transforms $\mathrm{k} = 2,...,\mathrm{c}_{\mathrm{l}+1}$.

Figure 5. Our example network for the wavelet CNN. The network consists of three residual blocks, indicated by the connecting arrows, an SPR to upsample the 2D space followed by two non-linear layers before output without nonlinearity.

Figure 6. Simple illustration of the SPR operation using four input features to output two features by periodic resampling.

Figure 7. Barents Sea water bottom modeled from bathymetry data from side-scan sonar and diffraction lines used in the modelling, defined on a grid size of $6.25 \times 6.25\ \mathrm{m}$. The water depth varies from 325 – 360m while the regular grid of perpendicular diffraction lines (in blue) are located at a constant depth of 518m. Red rectangle depicts the modelling area.

Figure 8. Zoomed section of bin locations, position and grid size of modelled constant offset and zero-azimuth data. **a**) ground truth and **b**) after decimation with three of four traces removed in the crossline direction.

Figure 9. Zoomed section of the diffraction modelled data displayed in the inline direction where a) is the ground truth and b) is the subsampled counterpart.

Figure 10. This figure shows reconstruction results in the crossline direction from the shallow section of the wavefield, i.e. the water bottom. From **a**) the ground truth we observe a complex wavefield setting, which follows that **b**) the subsampled crossline direction yields a challenging wavefield to reconstruct. From **c**) the 3D Fourier and **e**) the wavelet CNN, we observe more residual energy within the **d**) 3D Fourier interpolation error than the **f**) wavelet CNN reconstruction error.

Figure 11. This figure shows reconstruction results in the crossline direction from the deeper section of the wavefield, i.e. the diffraction lines. From **a**) the ground truth we see a much simpler wavefield setting than the water bottom in Figure 10, which follows that **b**) the subsampled crossline direction gives a simpler pattern to reconstruct. From **c**) the 3D Fourier and **e**) the wavelet CNN, we observe more residual energy within the **d**) 3D Fourier interpolation error than the **f**) wavelet CNN reconstruction error.

Figure 12. This figure shows the migration results from a crossline section and a time slice through 450 ms, where **a**) and **b**) represents the migration result of the ground truth, **c**) and **d**) represents the migration result of the 3D Fourier interpolation and **e**) and **f**) represents the migration result of the wavelet CNN reconstruction.

Figure 13. An illustration of a single training example **a**) and **b**) in the shot domain and their corresponding frequency-wavenumber plot in **c**) and **d**) respectively.

Figure 14. Reconstruction result from the inline direction from one arbitrary example in the test set where **a**) is the target gather, **b**) the subsampled, **c**) the

reconstruction, **d**) the reconstruction error, i.e. difference between **a**) and **c**), and **e**) the RMS amplitude spectrum of **a**), **b**) and **c**).

Figure 15. Reconstruction result displayed in the frequency-wavenumber domain from the test set presented in Figure 14 where **a**) is the target gather, **b**) is the subsampled, **c**) is the reconstruction and **d**) is the reconstruction error, i.e. difference between **a**) and **c**).

Figure 16. Reconstruction result in the crossline direction from an arbitrary shot gather extracted from the training data, where **a**) the input from the 3D shot gather with 14 traces and **b**) the reconstruction result along with **e**) the RMS-amplitude spectrum and **c**), **d**) their respective frequency-wavenumber spectra.

Figure 17. Timeslice view of the input and reconstructed 3D shot gather displayed in Figure 16, where **a**), **b**), **c**) and **d**) are the inputs with 14 cables and **e**), **f**), **g**) and **h**) are the reconstruction results output with 53 cables.

Figure A-1. The L_1 loss during training up to early stopping for the training set batches and the validation set on the field data.

Figure B-1. Scatterplot of the RMS error and PSNR of all the shots in the test set.

Figure C-1. The DWT of the target data and predicted DWT from the test set displayed in Figure 14.

LIST OF TABLES

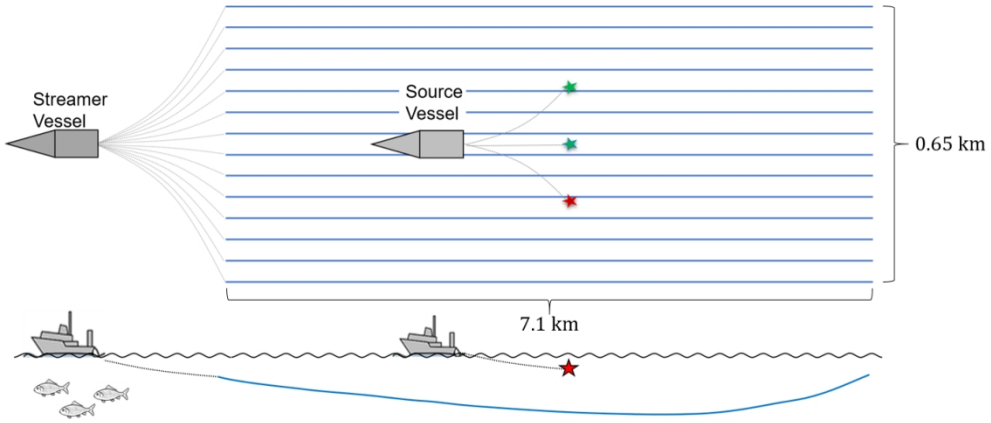

Figure 1. An illustration of a split-spread design employing both a streamer vessel source vessel with three sources.

89x39mm (400 x 400 DPI)

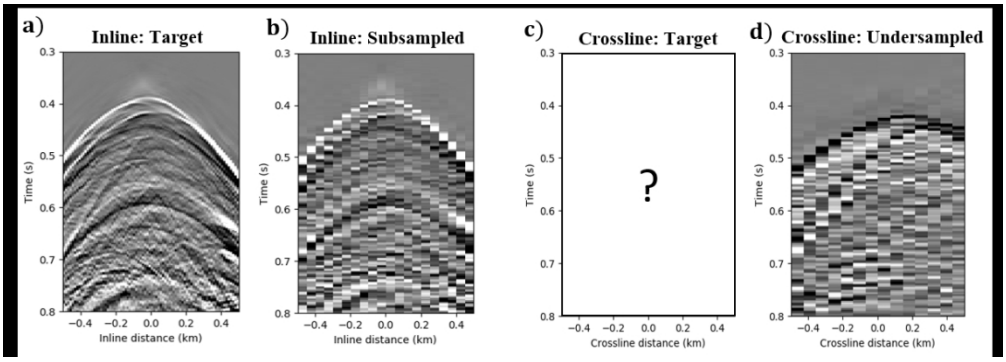

Figure 2. An illustration of inline training data and the crossline reconstruction problem. The two panels to the left, a) and b), shows a single training example in the shot domain where a) the target in the inline direction is subsampled to give b) the crossline representation. The inline targets and the subsampled counterparts in the training data are then used to learn the function f(·), in order to reconstruct c) the crossline target from d) the undersampled crosslines using the learned function f (·).

207x73mm (150 x 150 DPI)

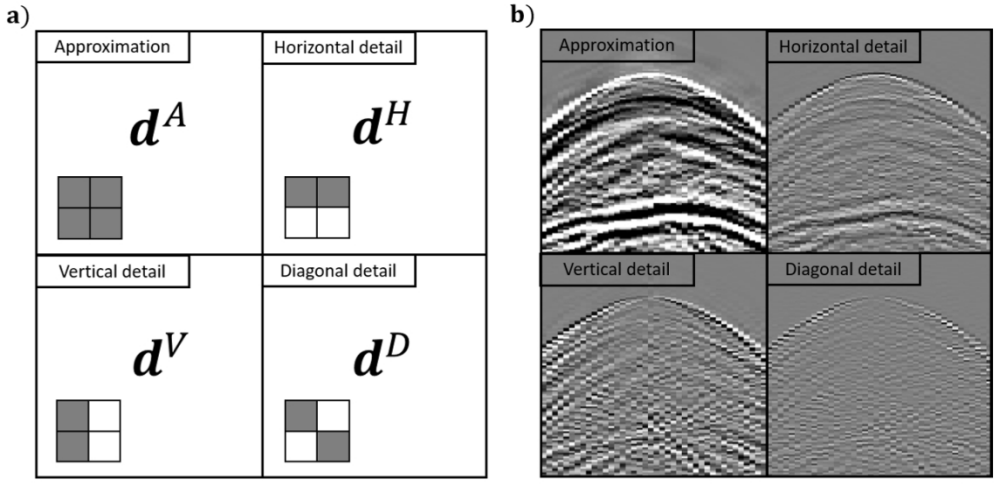

Figure 3. A simplified illustration of the wavelet decomposition, where a) is a schematic representation of the four-band split with their respective convolution kernels (gray=1/2, white=-1/2) and b) an example where a seismic source gather is decomposed using the Haar filter.

82x40mm (400 x 400 DPI)

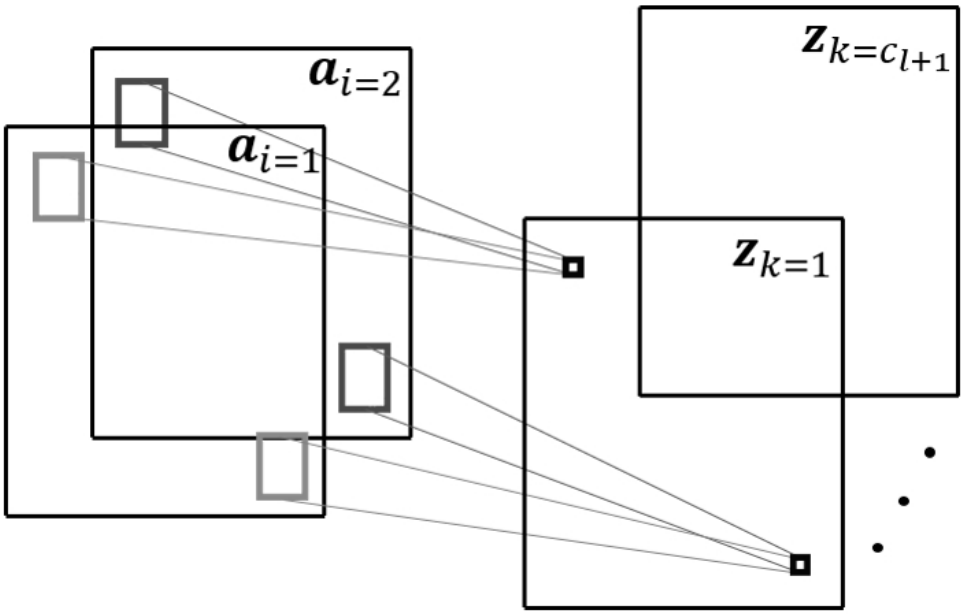

Figure 4. Simplified illustration of the convolution operation on an input with two features a_1and a_2 being convolved with a 2D kernel and superimposed with the bias parameter, to give the linear transform z_1. A different set of 2D kernels and biases are then used to compute the linear transforms k=2,…,c_(l+1).

46x29mm (400 x 400 DPI)

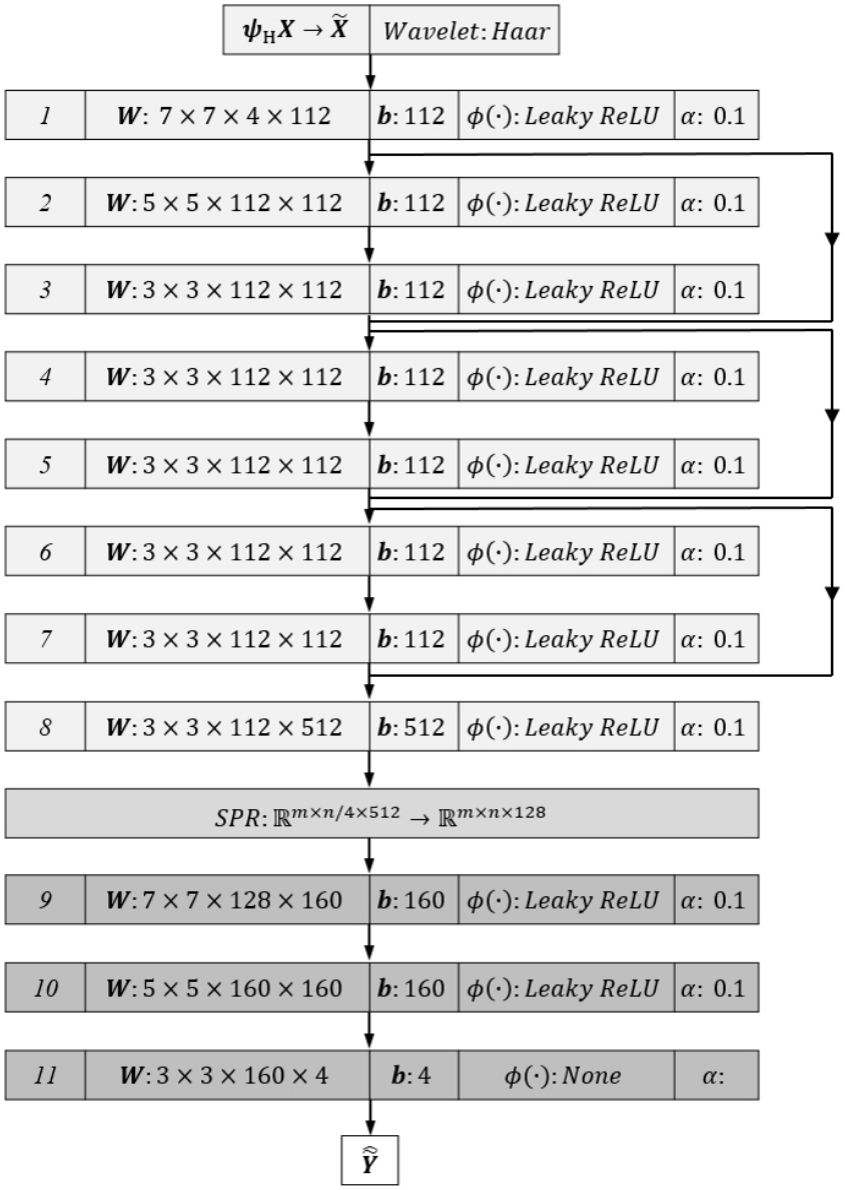

Figure 5. Our example network for the wavelet CNN. The network consists of three residual blocks, indicated by the connecting arrows, an SPR to upsample the 2D space followed by two non-linear layers before output without nonlinearity.

44x62mm (400 x 400 DPI)

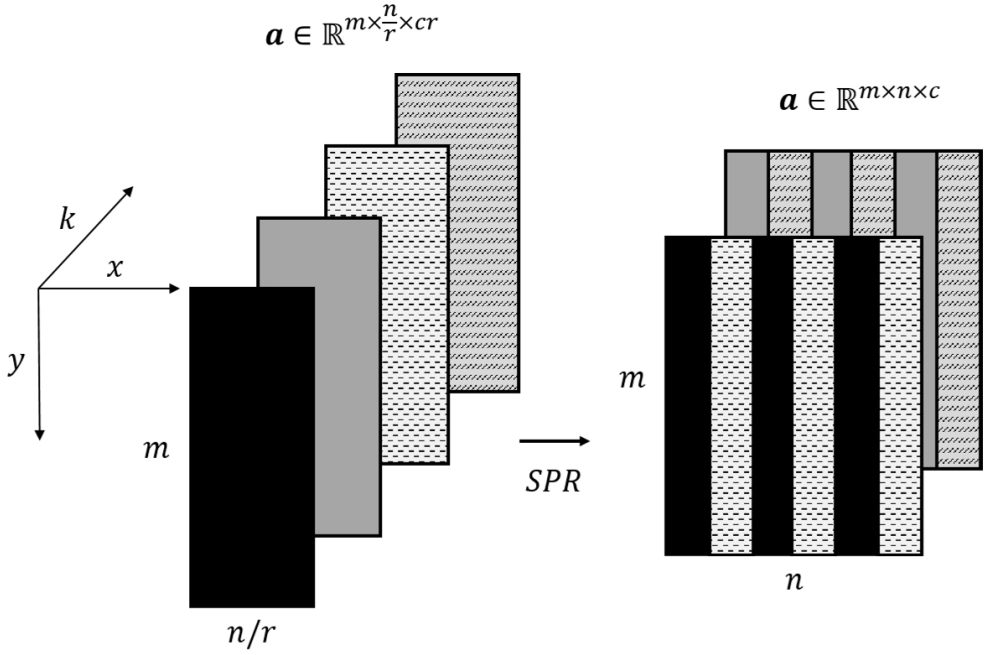

Figure 6. Simple illustration of the SPR operation using four input features to output two features by periodic resampling.

75x50mm (400 x 400 DPI)

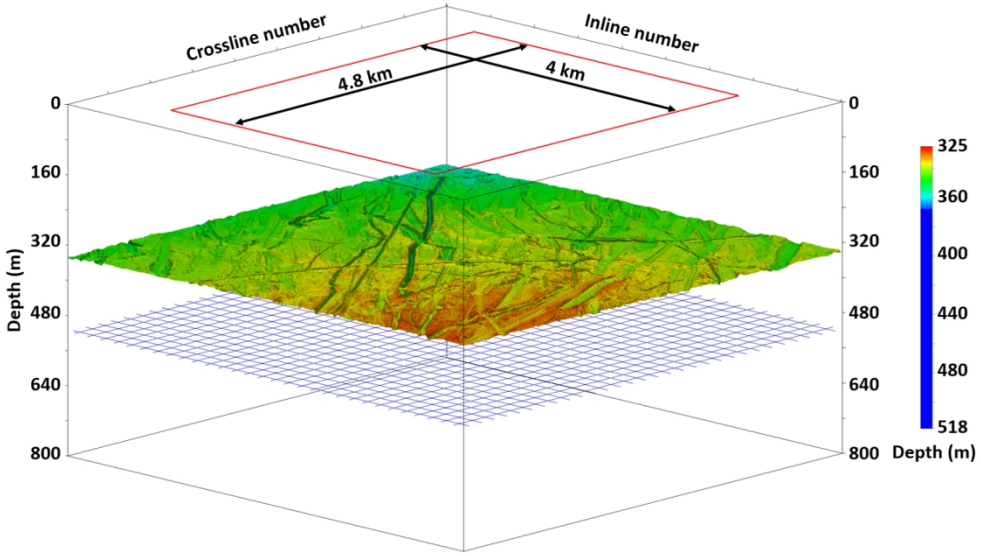

Figure 7. Barents Sea water bottom modeled from bathymetry data from side-scan sonar and diffraction lines used in the modelling, defined on a grid size of 6.25×6.25 m. The water depth varies from 325 – 360m while the regular grid of perpendicular diffraction lines (in blue) are located at a constant depth of 518m. Red rectangle depicts the modelling area.

104x60mm (400 x 400 DPI)

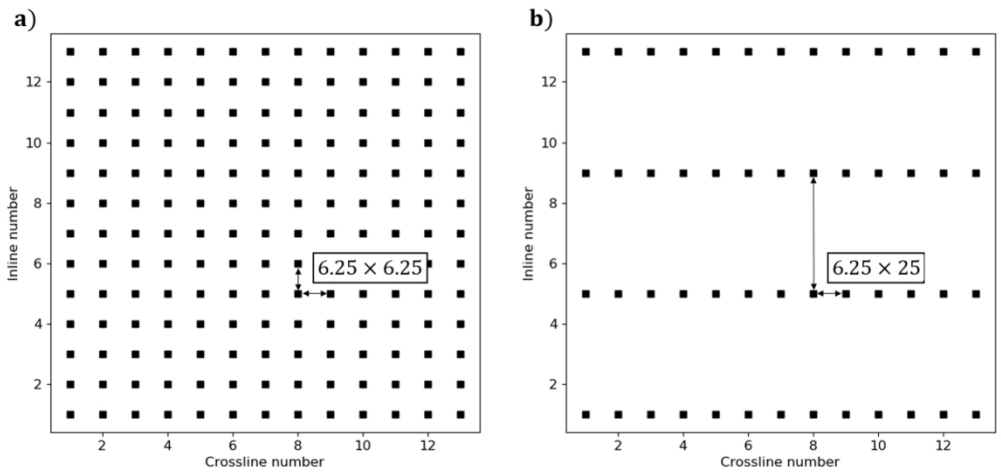

Figure 8. Zoomed section of bin locations, position and grid size of modelled constant offset and zero-azimuth data. a) ground truth and b) after decimation with three of four traces removed in the crossline direction.

85x40mm (400 x 400 DPI)

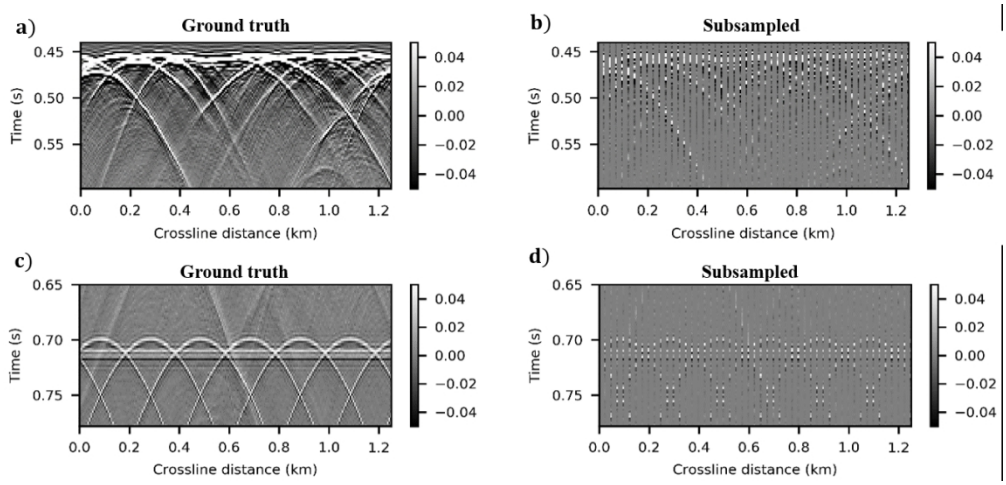

Figure 9. Zoomed section of the diffraction modelled data displayed in the inline direction where a) is the ground truth and b) is the subsampled counterpart.

227x108mm (150 x 150 DPI)

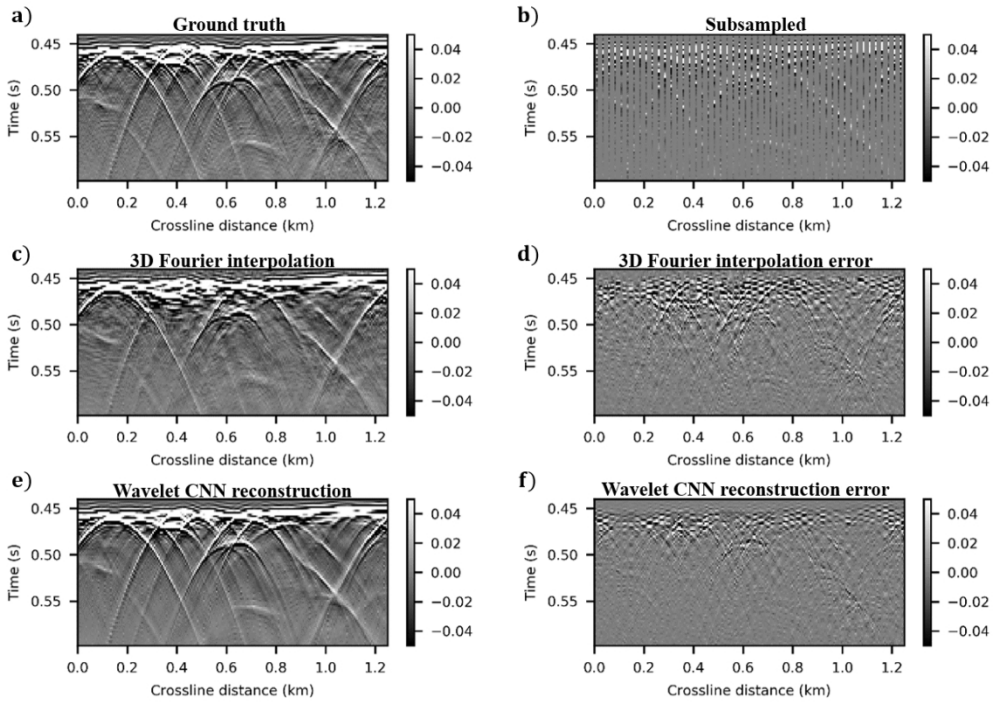

Figure 10. This figure shows reconstruction results in the crossline direction from the shallow section of the wavefield, i.e. the water bottom. From a) the ground truth we observe a complex wavefield setting, which follows that b) the subsampled crossline direction yields a challenging wavefield to reconstruct. From c) the 3D Fourier and e) the wavelet CNN, we observe more residual energy within the d) 3D Fourier interpolation error than the f) wavelet CNN reconstruction error.

218x154mm (150 x 150 DPI)

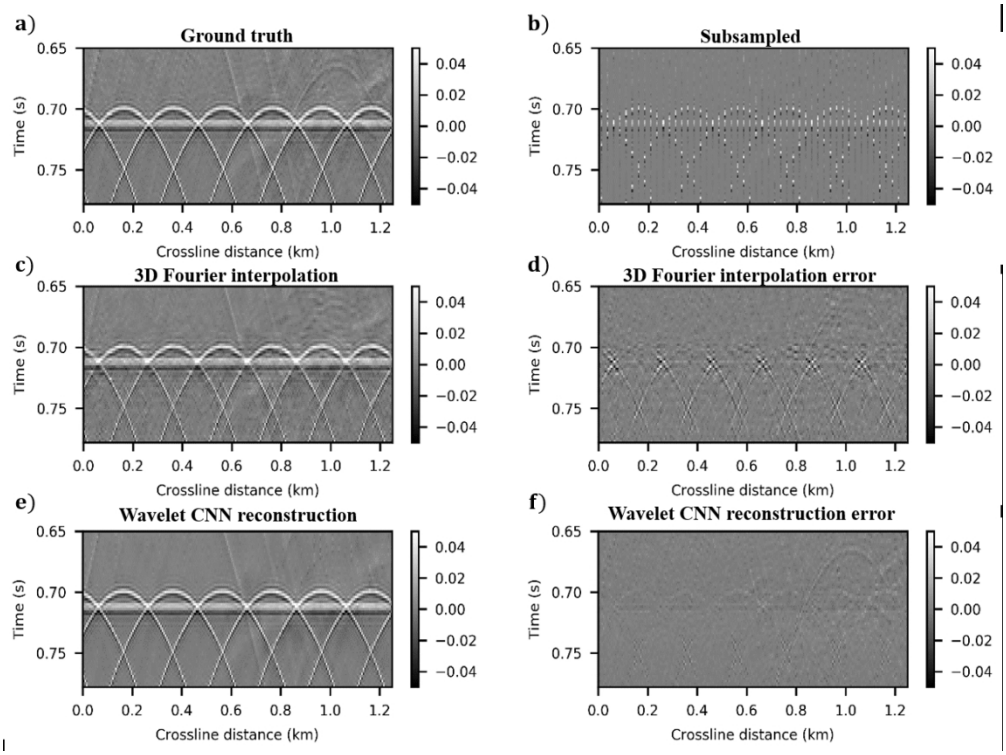

Figure 11. This figure shows reconstruction results in the crossline direction from the deeper section of the wavefield, i.e. the diffraction lines. From a) the ground truth we see a much simpler wavefield setting than the water bottom in Figure 10, which follows that b) the subsampled crossline direction gives a simpler pattern to reconstruct. From c) the 3D Fourier and e) the wavelet CNN, we observe more residual energy within the d) 3D Fourier interpolation error than the f) wavelet CNN reconstruction error.

218x163mm (150 x 150 DPI)

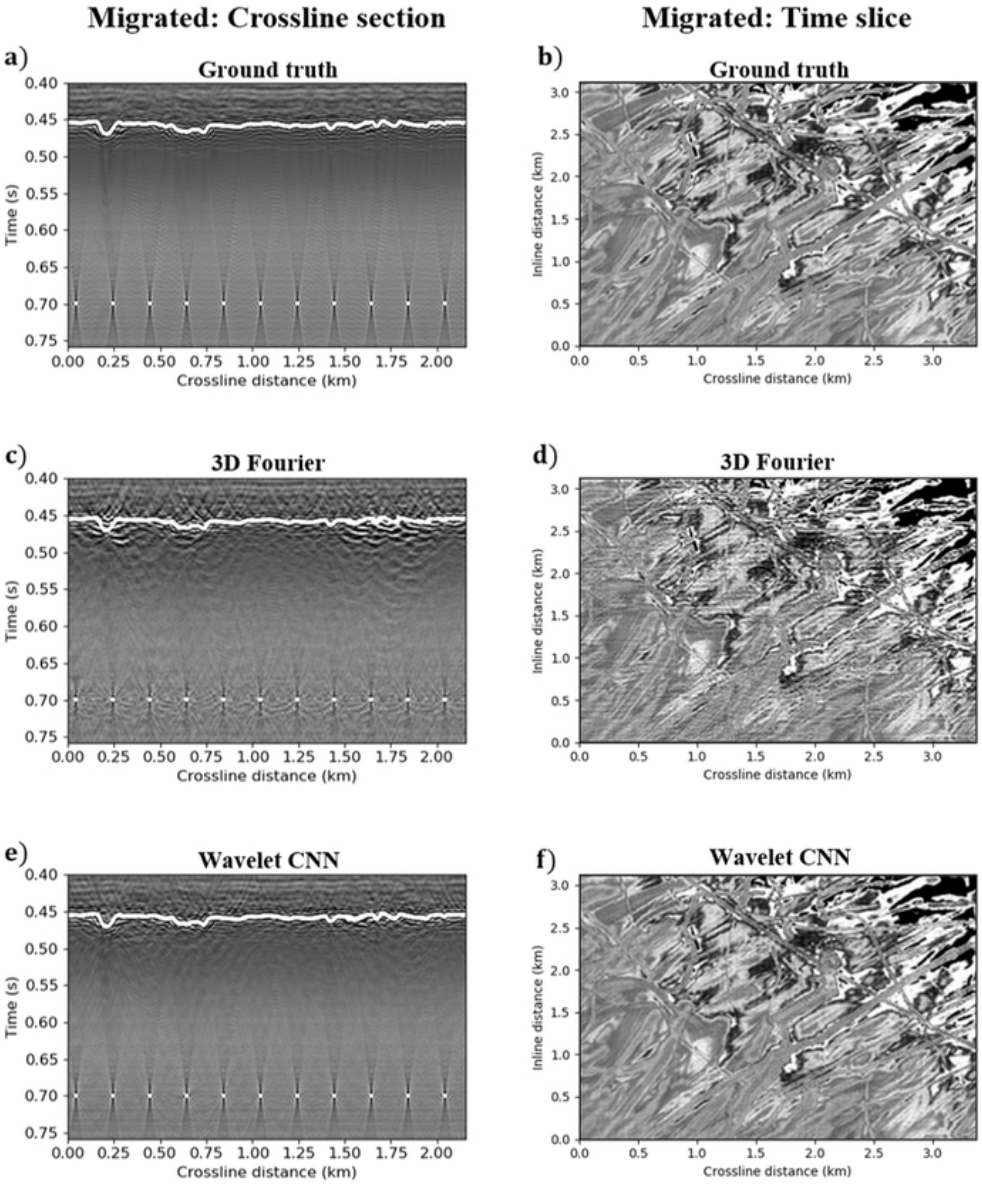

Figure 12. This figure shows the migration results from a crossline section and a time slice through 450 ms, where a) and b) represents the migration result of the ground truth, c) and d) represents the migration result of the 3D Fourier interpolation and e) and f) represents the migration result of the wavelet CNN reconstruction.

59x71mm (300 x 300 DPI)

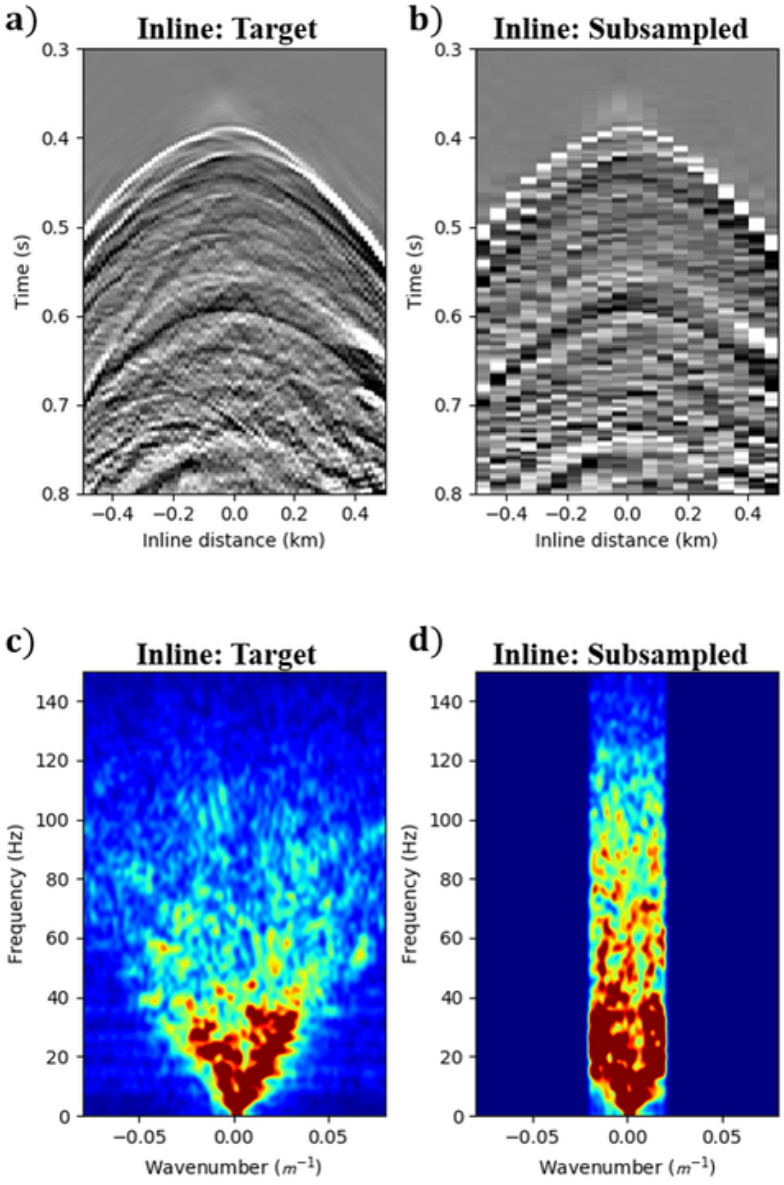

Figure 13. An illustration of a single training example a) and b) in the shot domain and their corresponding frequency-wavenumber plot in c) and d) respectively.

41x61mm (300 x 300 DPI)

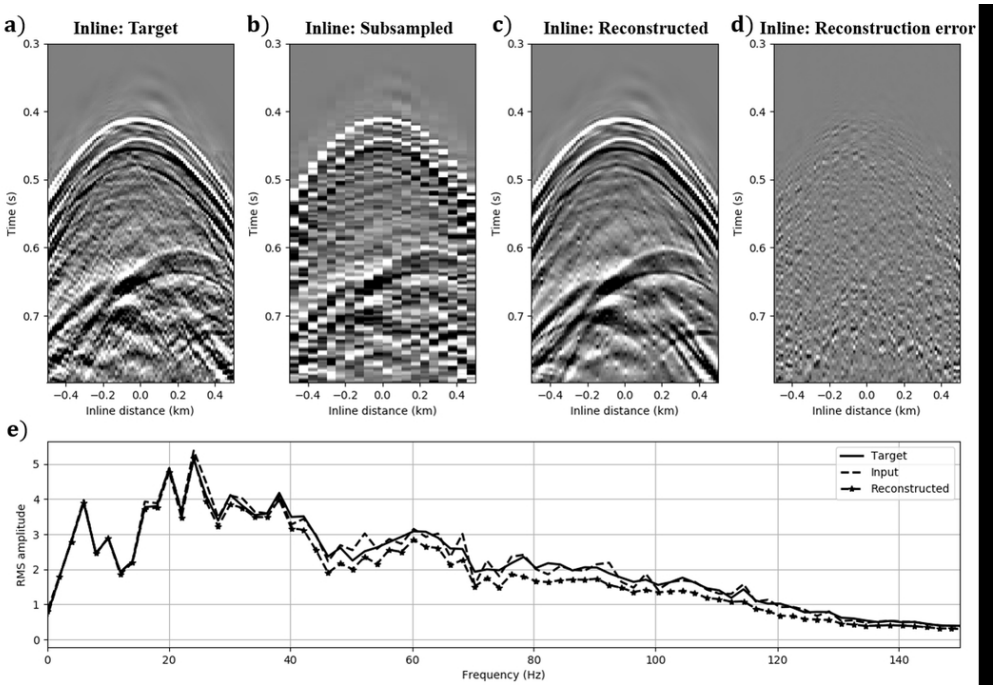

Figure 14. Reconstruction result from the inline direction from one arbitrary example in the test set where a) is the target gather, b) the subsampled, c) the reconstruction, d) the reconstruction error, i.e. difference between a) and c), and e) the RMS amplitude spectrum of a), b) and c).

88x60mm (300 x 300 DPI)

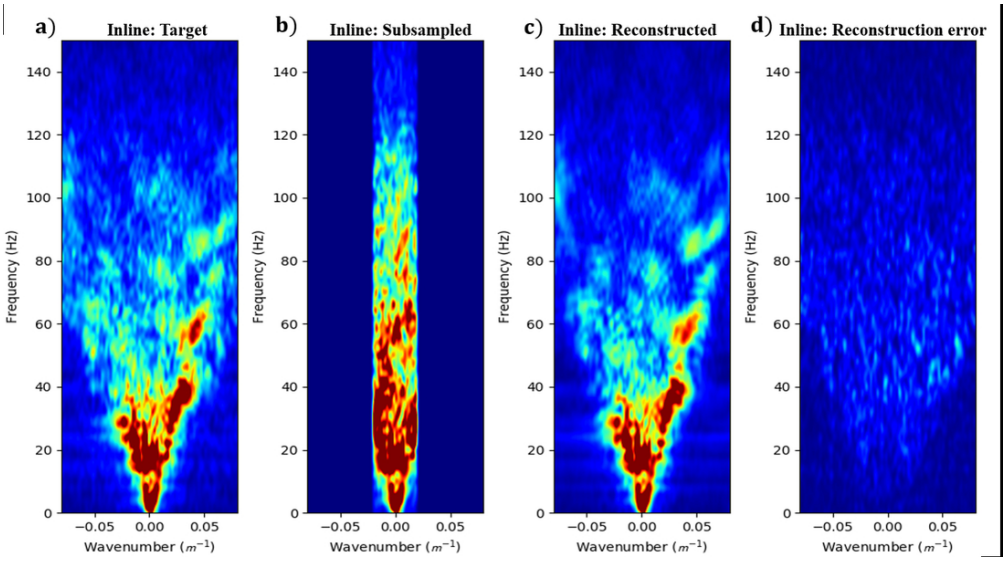

Figure 15. Reconstruction result displayed in the frequency-wavenumber domain from the test set presented in Figure 14 where a) is the target gather, b) is the subsampled, c) is the reconstruction and d) is the reconstruction error, i.e. difference between a) and c).

92x51mm (300 x 300 DPI)

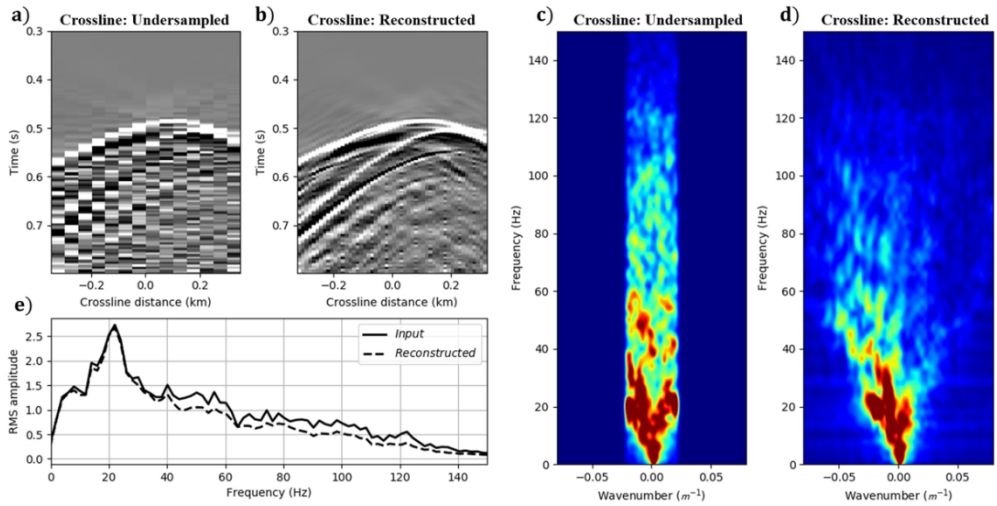

Figure 16. Reconstruction result in the crossline direction from an arbitrary shot gather extracted from the training data, where a) the input from the 3D shot gather with 14 traces and b) the reconstruction result along with e) the RMS-amplitude spectrum and c), d) their respective frequency-wavenumber spectra.

104x53mm (300 x 300 DPI)

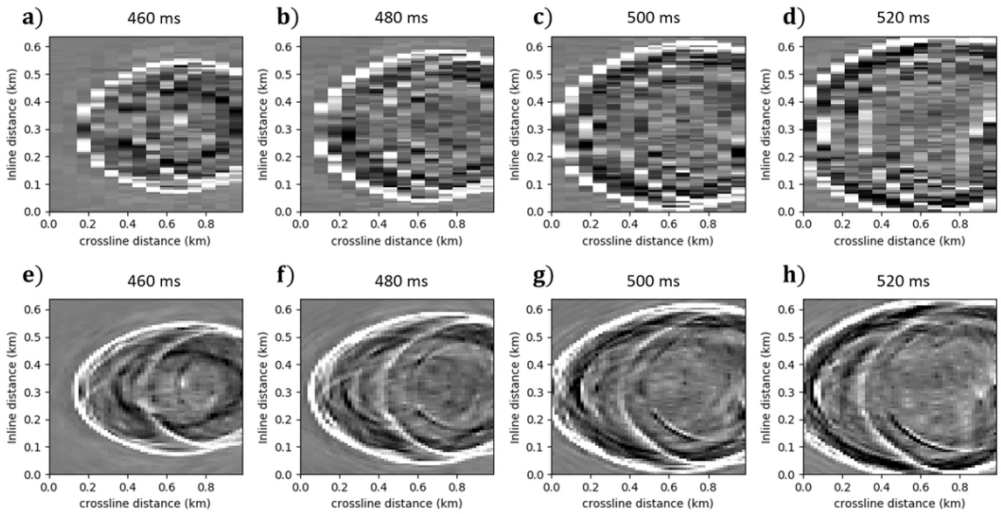

Figure 17. Timeslice view of the input and reconstructed 3D shot gather displayed in Figure 16, where a), b), c) and d) are the inputs with 14 cables and e), f), g) and h) are the reconstruction results output with 53 cables.

101x52mm (300 x 300 DPI)

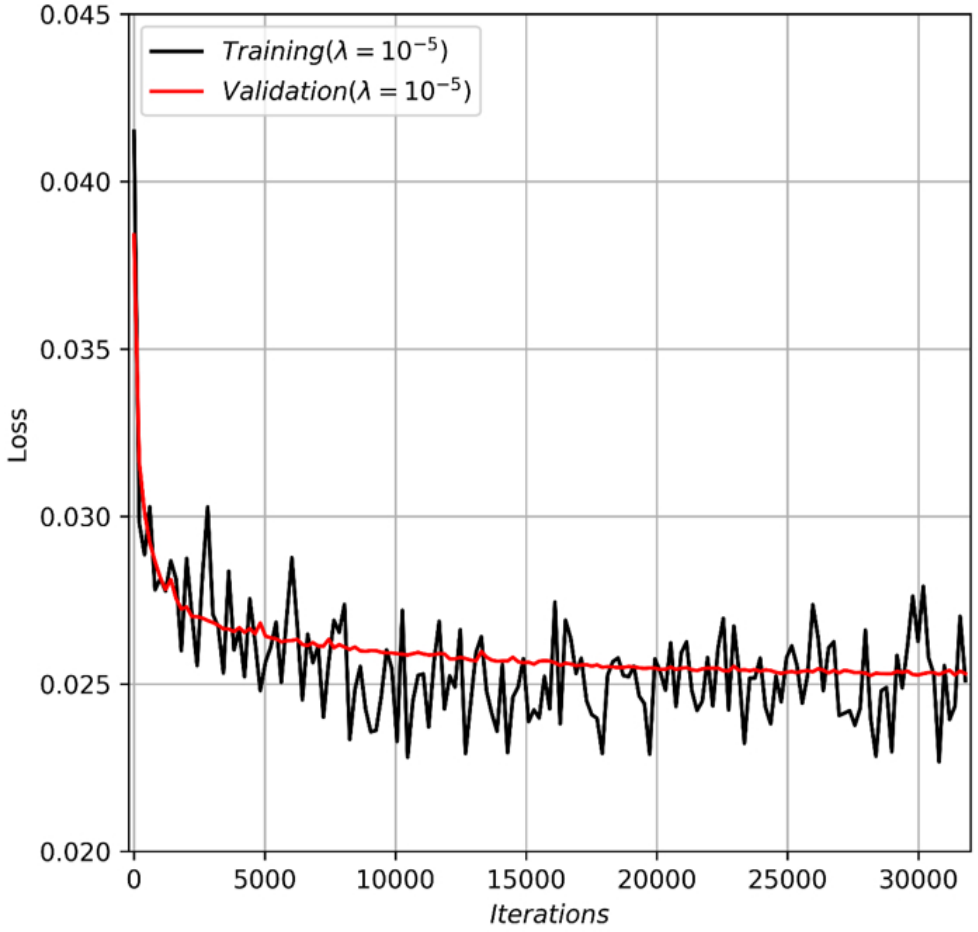

Figure A-1. The L_1 loss during training up to early stopping for the training set batches and the validation set on the field data.

120x113mm (150 x 150 DPI)

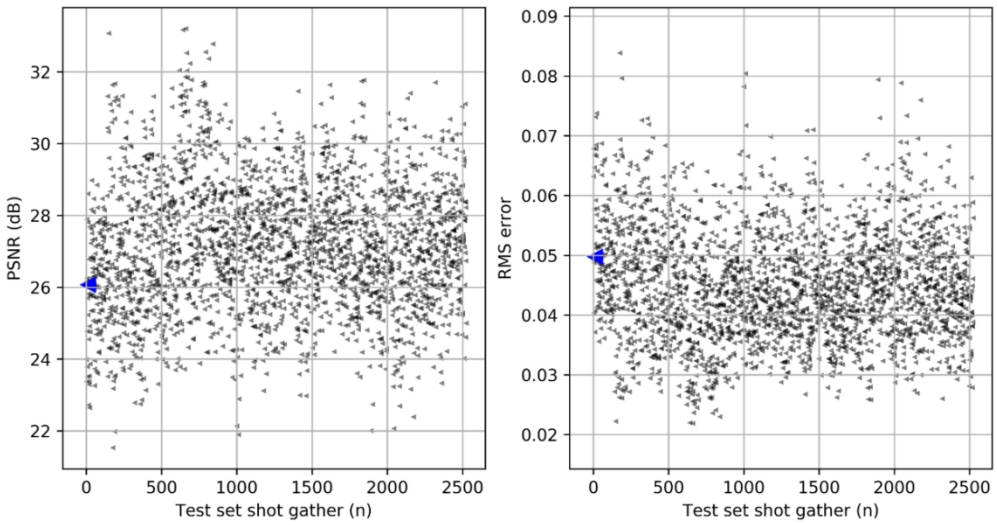

Figure B-1. Scatterplot of the RMS error and PSNR of all the shots in the test set.

211x110mm (150 x 150 DPI)

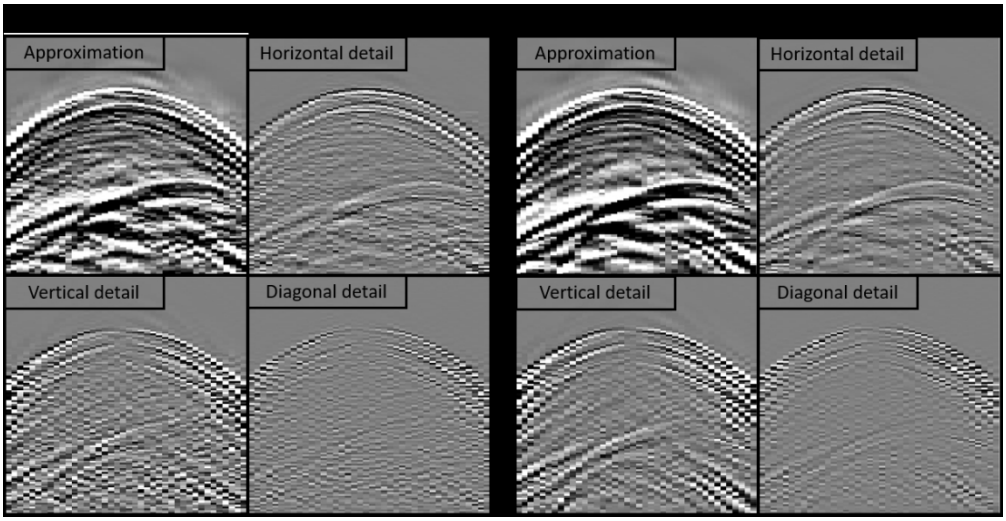

Figure C-1. The DWT of the target data and predicted DWT from the test set displayed in Figure 14.

210x108mm (150 x 150 DPI)

Table 1. RMS error and PSNR from the pre and post migrated crossline section from the entire interval.

	3D Fourier		Wavelet CNN	
	RMS	PSNR	RMS	PSNR
Pre migration	0.0086	41.3434	0.0047	46.4906
Post migration	0.0124	38.1423	0.0068	43.3990

Table 1. Average RMS error and PSNR comparison of the different sets.

Test set		Validation set		Training set	
RMS	PSNR	RMS	PSNR	RMS	PSNR
0.04484	27.1471	0.04487	27.1318	0.04418	27.2670