Surveys in Geophysics (2021) 42:69–95
https://doi.org/10.1007/s10712-020-09615-3

Deep Learning for Extracting Dispersion Curves

Tianyu Dai[1] · Jianghai Xia[2] · Ling Ning[2] · Chaoqiang Xi[2] · Ya Liu[2] · Huaixue Xing[3]

Received: 22 January 2020 / Accepted: 25 August 2020 / Published online: 6 September 2020

Abstract
High-frequency surface-wave methods have been widely used for surveying near-surface shear-wave velocities. A key step in high-frequency surface-wave methods is to acquire dispersion curves in the frequency–velocity domain. The traditional way to acquire the dispersion curves is to identify the dispersion energy and manually pick phase velocities by following energy peaks at different frequencies. A large number of dispersion curves need to be extracted for inversion, especially for surveys with long two-dimensional sections or large three-dimensional (3D) coverages. Human–machine interaction-based dispersion curves extraction, however, is still common, which is time-consuming. We developed a deep learning model, termed Dispersion Curves Network (DCNet), that can rapidly extract dispersion curves from dispersion images by treating dispersion curves extraction as an instance segmentation task. The accuracy of the dispersion curves extracted by our DCNet model is demonstrated by theoretical data. We used a 3D field application of ambient seismic noise to demonstrate the effectiveness and robustness of our method. The real-world results showed that the accuracy of the dispersion curves extracted from the field data using our method can achieve human-level performance and our method can meet the requirement of geoengineering surveys in rapidly extracting massive dispersion curves of surface waves.

Keywords Surface waves · Dispersion curves · Deep learning · Convolutional networks

1 Introduction

High-frequency surface-wave methods (Xia et al. 1999, 2002) have been widely used for near-surface shear (S)-wave velocity survey among active (e.g., Xia et al. 2003, 2012; Xia 2014; Ivanov et al. 2006; Luo et al. 2007; Socco et al. 2010; Foti et al. 2011; Pan et al.

✉ Jianghai Xia
jianghai_xia@yahoo.com; jhxia@zju.edu.cn

[1] Hubei Subsurface Multi-scale Imaging Key Laboratory, Institute of Geophysics and Geomatics, China University of Geosciences, 388 Lumo Rd., Wuhan 430074, Hubei, China

[2] Key Laboratory of Geoscience Big Data and Deep Resource of Zhejiang Province, School of Earth Sciences, Zhejiang University, 38 Zheda Road, Hangzhou 310027, Zhejiang, China

[3] Nanjing Center of the Geological Survey, China Geological Survey, 534 Zhongshan East Road, Nanjing 210016, Jiangsu, China

2016a; Zhang and Alkhalifah 2019a) and passive seismic investigations (e.g., Louie 2001; Okada 2003; Park and Miller 2008; Cheng et al. 2015, 2016; Zhang et al. 2020). By using dispersion imaging methods, such as the τ–p transformation (McMechan and Yedlin 1981), the F–K transformation (Yilmaz 1987), the phase shift (Park et al. 1998), the frequency decomposition and slant stacking (Xia et al. 2007), and the high-resolution linear Radon transformation (HRLRT) (Luo et al. 2008), dispersion images can be easily transformed from shot gathers. A one-dimensional (1D) S-wave velocity can be achieved by inverting the dispersion curves (Xia et al. 1999; Socco and Boiero 2008; Boaga et al. 2011) picked from each dispersion image. Based on the 1D approximation, each inverted model of dispersion curves represents the mean S-wave velocity of the underground structure (Boiero and Socco 2010). Two-dimensional (2D) S-wave velocity profile or three-dimensional (3D) S-wave velocity model can be constructed by combining multiple 1D results. The key step to ensure the accuracy of inversion is to acquire the reliable dispersion curves of surface waves in the frequency–velocity (f–v) domain (Shen et al. 2015). The most traditional way to acquire the dispersion curves is to identify the dispersion energy from dispersion image and manually pick phase velocities by following peaks at different frequencies. At present, the human–machine interaction method can achieve semi-automatically extraction of dispersion curves by manually clicking at dispersion energy areas (Shen 2014); however, manual identification of different dispersion energy modes is still indispensable.

With the widespread use of high-frequency surface-wave methods and the increasing number of seismic data we need to deal with, people are reluctant to spend much time on duplicating tasks of acquiring dispersion curves. The common tasks are: 1. generating 2D S-wave velocity profiles by aligning a large number of 1D S-wave velocity models (Bohlen et al. 2004; Yin et al. 2016; Mi et al. 2017; Pan et al. 2019), 2. delineating shallow S-wave velocity structure using multiple ambient-noise surface-wave methods (Pan et al. 2016b), and 3. estimating for a 3D S-wave velocity model (Pilz et al. 2013; Ikeda and Tsuji 2015; Pan et al. 2018; Mi et al. 2020). Meanwhile, there is a newly developed seismic data acquisition technology—Distributed Acoustic Sensing (DAS) (Daley et al. 2013; Ning and Sava 2018; Song et al. 2019). Innumerable amounts of data can be acquired by this new technology. There is no doubt that manually picking dispersion curves will be unrealistic in the near future. In addition, the dispersion curves of manual extraction have certain subjectivity. Some significant energy (i.e., spatial aliasing and "crossed" artifacts) (Dai et al. 2018a; Cheng et al. 2018a) confuse people when picking dispersion curves because each person has different experiences in the surface-wave data processing.

In recent years, as a new research direction, data-driven deep learning has aroused the interest of geophysicists. Deep learning models, building a layered architecture similar to the human brain, can extract the features from the bottom to the top of the input data layer by layer, thus establishing a good mapping between signal and semantic (LeCun et al. 2015). With the rapid development of deep learning, more powerful computing, and the increasing data processing capacity, recent researches have applied deep learning to geophysics. Perol et al. (2018) presented a convolutional neural network for earthquake detection and location. Zachary et al. (2018) trained a convolutional neural network (CNN) to detect seismic body wave phases. Zhang et al. (2018) applied CNN to predict seismic lithology. Mao et al. (2019) developed a CNN to predict the subsurface velocity information. Wang and Chen (2019) used residual learning of deep CNN for seismic random noise attenuation. Wang et al. (2019) developed a deep learning model to automatically pick a great number of first P- and S-wave arrival times precisely from local earthquake seismograms. Wu et al. (2019) proposed a multiple-task learning CNN to simultaneously perform three seismic image processing tasks of detecting faults, structure-oriented smoothing with edge preserving, and estimating seismic local structure orientations. Yang and Ma (2019) proposed a deep fully convolutional neural network for velocity model building directly from raw seismograms. In addition, within the framework of full waveform inversion, Ovcharenko et al. (2019) used deep learning to extrapolate data at low frequencies, and Zhang and Alkhalifah (2019b) utilized deep neural networks to estimate the distribution of facies in the subsurface as constraints. Few studies (e.g., Dai et al. 2018b), however, have applied deep learning to the surface-wave dispersion curves extraction. It is imperative to adopt deep learning technology for tasks with large volumes of data and tedious work. It often leads to greater efficiency of geophysical data processing, reduces cost, and reduces biases associated with human–machine interaction influenced by past professional experience.

Dai et al. (2018b) discussed that dispersion curves extraction can be regarded as a semantic segmentation (Long et al. 2015; Badrinarayanan et al. 2017) task. For multimode or more complex dispersion images, however, the concept was not suitable. The generated binary segmentation still needs to be separated into the different surface-wave modes, especially when the convergence of dispersion energy is insufficient thereby causing the energy of different surface-wave modes to be very close to each other. Inspired by the success of instance segmentation which has been well studied in computer vision via deep learning models (e.g., Dai et al. 2015; Romera-Paredes and Torr 2016; Bai and Urtasun 2017; Ren and Zemel 2017; He et al. 2017; Neven et al. 2018; Chen et al. 2020), we regard the dispersion curves extraction as an instance segmentation task.

In this paper, we adopt the ideas from these previous studies and develop a deep learning model, called Dispersion Curves Network (DCNet), to extract dispersion curves in the *f*–*v* domain. DCNet is a multitask network model (e.g., Caruana 1997; Ruder 2017; Wu et al. 2019), which consists of a segmentation branch and an embedding branch. Learning multiple related tasks from data improves efficiency and prediction accuracy by exploiting commonalities and differences through the multiple tasks (Evgeniou and Pontil 2004). The segmentation branch segments the dispersion images into two classes, background and dispersion energy, while the embedding branch further distinguishes the segmented dispersion energy pixels into different mode instances. Surface-wave mode separation technique (Luo et al. 2009a) can be achieved, in which each mode of dispersion curve forms its own instance within the dispersion energy class. We design a data augmentation method for surface-wave energy recognition and create a data set with 25,000 labeled surface-wave dispersion energy data for training the DCNet model. We test the accuracy of the DCNet model extracted results by comparing them with the theoretical dispersion curves of simulated data. These tests indicate that our DCNet model is very effective with high accuracy. We also apply the DCNet model to a 3D passive surface-wave field data to automatically extract a large number of dispersion curves and generate a 3D S-wave velocity model by assembling the 1D S-wave velocity profiles inverted from each dispersion curve. The effectiveness and robustness of our method were demonstrated by comparing the 3D S-wave velocity model with the borehole data.

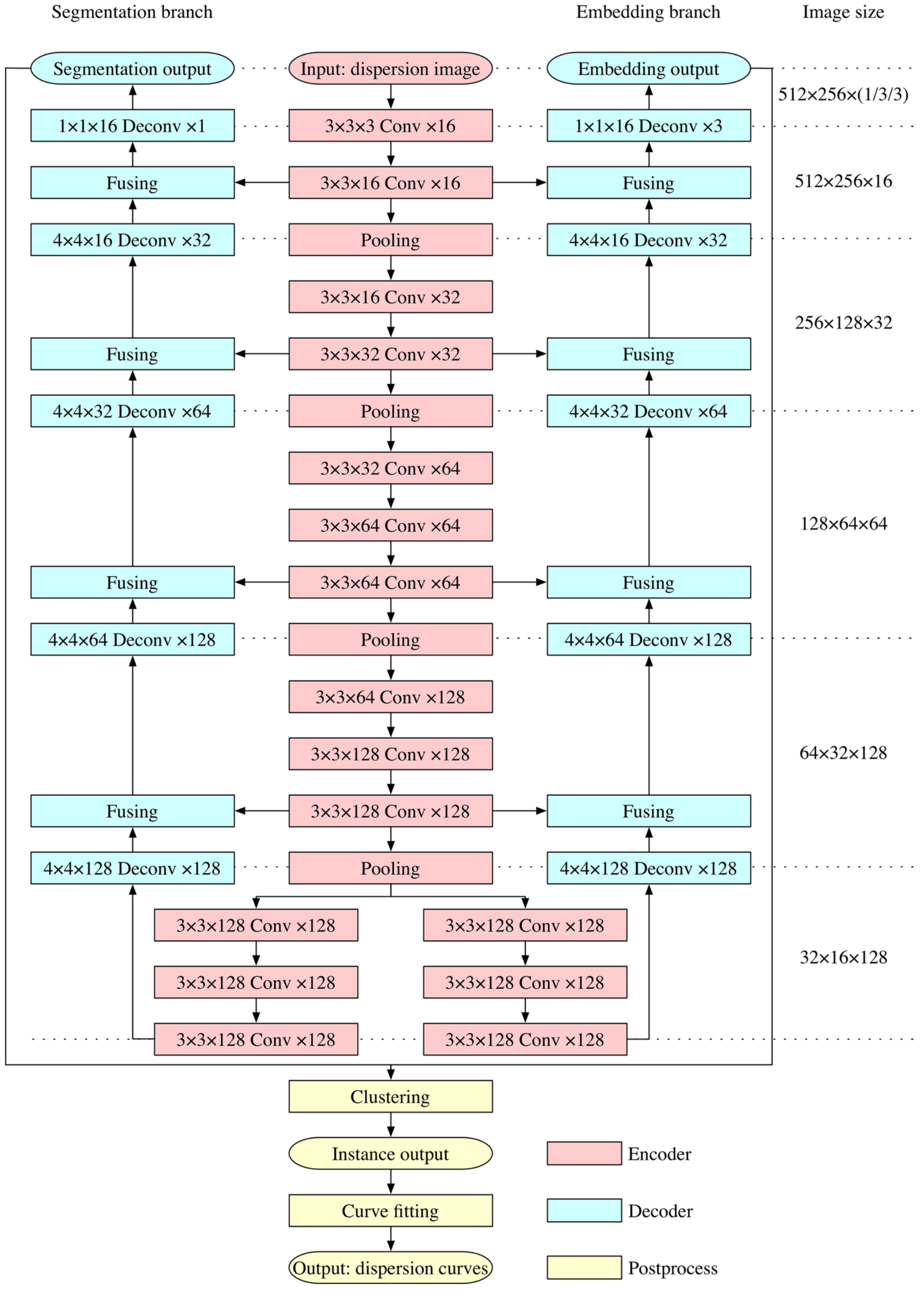
Fig. 1 The architecture of the DCNet. The end of each Pooling indicates the end of a stage of the encoder ▶ part in the middle of the architecture. The left and right parts show the segmentation and embedding branches, respectively. Each rounded block represents input or output. Each block corresponds to an operation. The abbreviations of operations are shown in the network architecture, i.e., Conv, Pooling, Deconv, Fusing, are defined in Table 1. Specifically, Conv and Deconv represent the convolution layer and deconvolution layer with batch normalization (BN) operation and rectified linear unit (ReLU) operation, respectively. The first three-dimensional parameter in the Conv and Deconv operation means the size of its convolutional kernels; the last parameter means the number of convolutional kernels; the strides of Conv is 1×1; the strides of Deconv is 2×2; the padding use "same", which means output filled with zeros is the same size with input. The kernel size of Pooling is 2×2. The image size for each input or output is shown between two dashed lines

2 Method and Experiments

2.1 Network Architecture

DCNet is trained end to end for surface-wave dispersion curves extraction, by regrading dispersion curves extraction as an instance segmentation task. In this way, the network can extract surface-wave multimode dispersion curves. DCNet is a multitask encoder–decoder network structure consisting of a segmentation branch and an embedding branch (Fig. 1). Formulas of operations defined in DCNet are listed in Table 1. The encoder part of DCNet is modified from the VGGNet (Simonyan and Zisserman 2014) network model, which has a simple architecture with convolution, pooling, and fully connected layers. It is mainly used for image classification tasks. The decoder part is designed based on the idea of the fully convolutional networks (Long et al. 2015).

DCNet shares the first four stages of encoder between the two branches, while the last stage of the encoder and the full decoder are each a separate branch. The segmentation branch is trained to segment the dispersion images into two classes, background and dispersion energy. To distinguish the dispersion energy pixels identified by the segmentation branch, we trained another branch of DCNet for dispersion energy embedding. The last layer of the segmentation branch outputs a one channel image, whereas the last layer of the embedding branch outputs a three-channel image. As shown in Fig. 1, each branch's loss term is equally weighted and back-propagated through the network. By using the discriminative loss function proposed by De Brabandere et al. (2017), the embedding branch is trained to output an embedding for each surface-wave mode. The discriminative loss function has a good performance on segmentation task of single class with multiple instances, and accepts any number of instances. The distance between pixel embeddings belonging to the same surface-wave mode is small, whereas the distance between pixel embeddings belonging to different surface-wave modes is maximized. After using the output of the segmentation branch (Fig. 2b) as a mask on the output of the embedding branch (Fig. 2c, d), the surface-wave embeddings (Fig. 2e, f) are clustered together and assigned to their mode cluster centers using DBSCAN (Ester et al. 1996) with the parameters of the radius $Eps = 0.5$ and the minimum number of neighboring points $MinPts = 50$ in this paper (Fig. 2g). Finally, an instance segmentation image (Fig. 2h) of surface waves with mode separation can be achieved, and then we can fit a curve through the corresponding peaks of phase velocity from each surface-wave mode instance to obtain the final extracted dispersion curves (Fig. 2i, j).



2.2 Datasets and Training

To ensure the training results cover a variety of scenarios, we used a large amount of labeled dispersion images (Fig. 3) for training our DCNet model. We used 200 real-world active source data, 1000 real-world passive source data, and 600 simulated data (the creation step

Table 1 Formulas of operations defined in DCNet, where * refers to a 4-dimensional convolution; K and $\bar{K}$ represent the convolutional kernels; b represents the bias; and *ksize* represents the kernel size; *input*1 and *input*2 represents the outputs of the Conv and Deconv at the same stage, respectively

Operation	Formula
Conv (convolution)	$output = K * input + b$
Deconv (deconvolution)	$output = \bar{K} * input + b$
BN (batch normalization)	$output = \frac{input - mean\left[input\right]}{\sqrt{Var\left[input\right]}}$
ReLU (rectified linear unit)	$output = max(0, input)$
Pooling (max-pooling)	$output = max\left[input\right]_{ksize}$
Fusing	$output = input1 + input2$

is shown in Table 2) to generate the dispersion images by using the common imaging methods, such as τ–p transformation (McMechan and Yedlin 1981), phase shift (Park et al. 1998), frequency decomposition and slant stacking (Xia et al. 2007), and HRLRT (Luo et al. 2008). They contain not only dispersion images with the fundamental mode but also dispersion images with fundamental and higher modes, or theoretical dispersion images with multimode. The data set contains real-world data with a lot of noise, spatial aliasing (Dai et al. 2018a) or "crossed" artifacts (Cheng et al. 2018a), and synthetic data with spatial aliasing or "crossed" artifacts, to ensure that the training result can cope with the existence of useless or fake energy. With a large number of real-world surface-wave data, we can simulate many theoretical data. However, it is still not enough to train a neural network only by manual data labeling. To solve this problem, we expanded the dataset through data augmentation, which can improve the robustness of the model and avoid overfitting. General data augmentation methods were not suitable for dispersion energy recognition, such as flipping, rotating, zooming, translating, and color jittering (Howard 2013). Therefore, we added different levels of noise to each (real-world or simulated) shot gather and generated dispersion images with different frequency ranges and velocity ranges. The imaging results are both normalized and non-normalized in the frequency domain (Fig. 4), which increases the number of samples. In this way, we obtained 25,000 samples in total, of which 80% were used for training and the rest for validation.

DCNet was trained by using TensorFlow libraries (Abadi et al. 2016). The dispersion images were rescaled to 512×256. We used the Adam (Kingma and Ba 2014) for optimizing the network parameters. The beginning learning rate was 0.0005 and linearly decays to its 1% with 36 epochs, and all the 20,000 training data sets were processed at each epoch. We initialize random weights and use a batch size of 8 considering a trade-off between generalization performance and memory limitation on a personal laptop with an NVIDIA GeForce RTX 2060 GPU.

The loss function of segmentation branch $\mathcal{L}_{seg}$ is,

$$\mathcal{L}_{seg} = \frac{\sum_{i=1}^{N} y_i' \log\left(\frac{e^{y_i}}{\sum_{i=1}^{N} e^{y_i}}\right) w_{class}}{N} \tag{1}$$

where $w_{class} = \frac{1}{\ln\left(\varepsilon + \frac{N_{class}}{N}\right)}$ is a bounded inverse class weighting (Paszke et al. 2016), because the dispersion energy area is always much smaller than the background area; N_{class} represents pixel number of the class (dispersion energy or background) in label and N represents the pixel number of the whole label; $\varepsilon = 1.02$ which is a control parameter that limits w_{class} to the interval of [1:50]. y_i and $y_i^{'}$ represent the prediction values and label values, respectively.

The loss function of embedding branch $\mathcal{L}_{emb}$ is,

$$\mathcal{L}_{var} = \frac{\sum_{c=1}^{C} \sum_{i=1}^{N_c} \left[\|m_c - x_i\| - \delta_v\right]_+^2}{CN_c}$$

$$\mathcal{L}_{dist} = \frac{\sum_{c_a=1}^{C} \sum_{c_b=1}^{C} \left[2\delta_d - \|m_{c_a} - m_{c_b}\|\right]_+^2}{C(C-1)}, \left(c_a \neq c_b\right)$$

$$\mathcal{L}_{reg} = \frac{\sum_{c=1}^{C} \|m_c\|}{C}$$

$$\mathcal{L}_{emb} = \alpha \cdot \mathcal{L}_{var} + \beta \cdot \mathcal{L}_{dist} + \gamma \cdot \mathcal{L}_{reg} \tag{2}$$

where $\mathcal{L}_{var}$ is a variance term, which applies a pull force on each embedding toward the mean embedding of a surface-wave mode and it only activates when an embedding is further than δ_v from its mode cluster center; $\mathcal{L}_{dist}$ is a distance term, which pushes the cluster centers of different surface-wave modes away from each other and it only activates when they are closer than $2\delta_d$ to each other; C is the number of surface-wave modes; N_c is the number of elements in mode cluster; x_i represents a pixel embedding and $m_c = \frac{1}{N} \sum_{i=1}^{N_c} x_i$ represents the mean embedding of the mode cluster; $\| \cdot \|$ is the L2 distance and $[x]_+ = \max(0, x)$; $\mathcal{L}_{reg}$ is a regularization term to keep the activations bounded, which applies a small pull force on all mode clusters toward the origin.

When $\delta_d > \delta_v$, each embedding is closer to its own mode cluster center than to any others. When $\delta_d > 2\delta_v$, each embedding is closer to all embeddings of its own mode cluster than to any embedding of a different mode cluster. Therefore, we set $\delta_v = 0.5$ and $\delta_d = 3$ to ensure that the embedding of each mode is far away from any other embedding of different modes, and keep $\mathcal{L}_{seg}$ and $\mathcal{L}_{emb}$ values on a comparable order of magnitude. We set $\mathcal{L}_{var}$ and $\mathcal{L}_{dist}$ to have the same weight and $\mathcal{L}_{reg}$ has a small weight with $\alpha = \beta = 1$ and $\gamma = 0.001$.

The accuracy is calculated as the average correct number of points per image:

$$accuracy = \frac{1}{M} \sum_{img=1}^{M} \frac{C_{img}}{T_{img}} \tag{3}$$

where C_{img} represents the number of correct points and T_{img} represents the number of ground-truth points, and M represents the number of images.

As shown in Fig. 5, the loss functions for training (the blue curves in Fig. 5a, b) and validation (the orange curves in Fig. 5a, b) converge to small values while the accuracy (the orange curves in Fig. 5c) gradually increases to 95% after 36 epochs. Figure 6 also shows that the first two epochs converged rapidly. After 36 epochs, the embeddings of different modes were gradually separated, and embeddings of the same mode were gathered together.

Fig. 2 The process of extracting dispersion curves by DCNet. **a** The input dispersion image. **b** Output of the segmentation branch (white areas represent dispersion energy and black areas represent the background). **c** Output of the embedding branch (a 2D representation). **d** Output of the embedding branch (a 3D representation). **e** Using the output of the segmentation branch as a mask on the output of the embedding branch (a 2D representation). **f** Using the output of the segmentation branch as a mask on the output of the embedding branch (a 3D representation). **g** Using DBSCAN (Ester et al. 1996) to cluster the embeddings together. **h** The instance output. **i** The dispersion curves extracted from the energy of each mode. **j** The final extraction result ▸

2.3 Theoretical Data Tests

In order to test the accuracy of the results of DCNet model, we simulated two typical models which parameters are shown in Table 3. The geophone interval was 1 m with the nearest offset of 10 m. The length of the record was 600 ms with a 0.1 ms sample interval. The shot gathers of Models 1 and 2 were simulated by a finite-difference method for modeling Rayleigh (Zeng et al. 2011) and Love waves (Luo et al. 2010), respectively. Their shot gathers and dispersion images generated by the phase shift (Park et al. 1998) are shown in Fig. 7. Model 1 was a common layered earth model with obvious fundamental mode dispersion energy (Fig. 7a, c) and Model 2 with stronger multimode surface-wave energy (Fig. 7b, d). We input their dispersion images into the DCNet model, and the processes of dispersion curves extraction are similar to Fig. 2. Their theoretical dispersion curves were calculated by Knopoff's method (Schwab and Knopoff 1972). Comparisons between dispersion curves extracted by DCNet and theoretically calculated by Knopoff's method are shown in Fig. 8a, b. The mean error of dispersion curves obtained by theoretical calculation and DCNet model is calculated by the following formula.

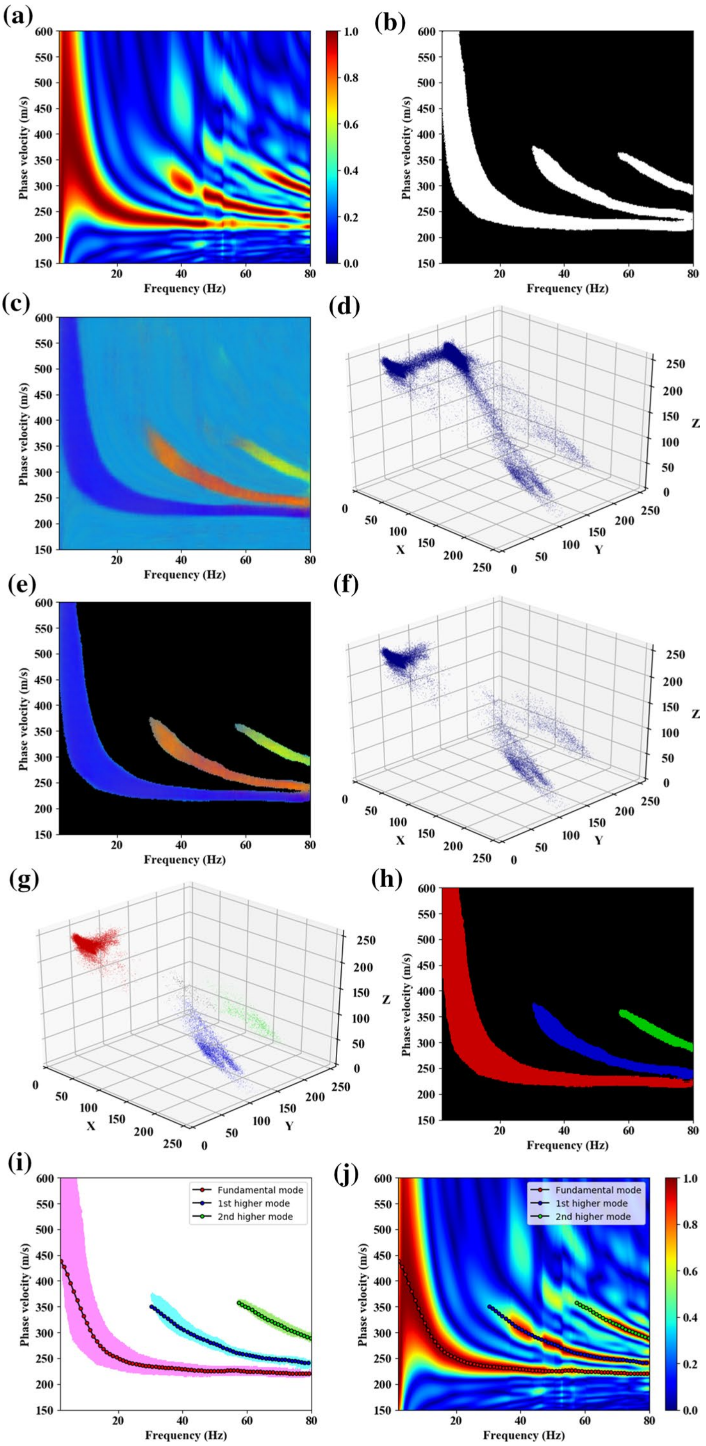
$$error = \frac{\sum_{i=1}^{N_f}\left(\left|\frac{T_{f_i}-M_{f_i}}{T_{f_i}}\right|\right)}{N_f} \tag{4}$$

where T_{f_i} and M_{f_i} represent the velocities of theoretical calculation and DCNet model at the i th frequency, respectively. N_f is the number of frequency points participating in comparison.

Calculated using Formula (4), the mean errors of the fundamental mode of surface waves (Fig. 8a) and the multimodes of surface waves (Fig. 8b) are 0.77% and 1.18%, respectively. Due to the errors expected in finite-difference simulation, the energy peaks at low-frequency (Fig. 8a) and high-mode (Fig. 8b) dispersion energy match the theoretical dispersion curves with relatively low accuracy. The comparisons demonstrated that our DCNet model was able to effectively pick accurate dispersion curves. Taking Model 2 as an example, as shown in Fig. 8f, the extracted results from dispersion images generated using different imaging methods (the phase shift, the τ-p transformation, the frequency decomposition and slant stacking, and the HRLRT) are almost the same. This indicates that the method we proposed has low sensitivity to different imaging methods.

2.4 Method Comparison

We compared our method with an automated extraction method of the fundamental mode dispersion curve (Taipodia et al. 2020) which is based on a threshold energy filtering of the dispersion image. We used the simulated data of Model 2 (Table 3) for a


(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
(i)
(j)
Phase velocity (m/s)
Frequency (Hz)
600
550
500
450
400
350
300
250
200
150
20
40
60
80
1.0
0.8
0.6
0.4
0.2
0.0
X
Y
Z
0
50
100
150
200
250
Fundamental mode
1st higher mode
2nd higher mode

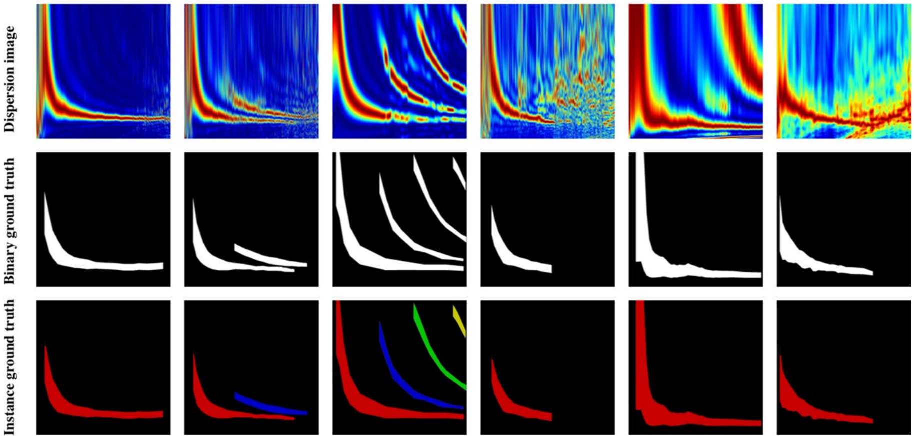


Fig. 3 Examples of dataset for training DCNet. Each column represents one sample. Each sample contains dispersion image, binary ground truth, and instance ground truth. The top row is the dispersion images, in which we hid the coordinate information that was not used during the training. The middle and bottom rows are the binary and instance ground truths, respectively. Columns 1–6, respectively, denote the common dispersion image, dispersion image with a higher mode, dispersion image with higher modes, dispersion image with a lot of noise, dispersion image with spatial aliasing, and dispersion image with "crossed" artifacts

Table 2 The steps of simulated data set creation

Step	Description
1	Randomly generate the number of receivers nx (12–36); Randomly generate the spacing interval dx (1–3 m)
2	Set the minimum offset $Xmin = 5 * dx$; Set the model size $xy = (nx * dx + Xmin) * 1.1 + 20$; The arrangement is set at the center of the model surface; Randomly set main frequency (15–30 Hz) Ricker wavelet
3	Use the random number η (0–1), set the thickness of the first layer $thks1 = dx * 1.2 + \eta_1 * 0.5$, and set the thickness of the second layer $thks2 = thks1 * 1.2 + \eta_2 * 0.5$, and so on, until the cumulative depth is greater than $nx * dx$; randomly initialize a Poisson's ratio σ (0.42–0.43); for each layer, use the corresponding η to initialize an increasing σ with $1 + \eta * 0.005$ times for each layer. This determines the number of layers, the thicknesses of each layer, and initial σ of each layer
4	Randomly initialize a V_s (150–300 m/s), and generate a 1.1–1.2 times random increment V_s model; generate a random sequence and perturb V_s and σ of the corresponding layer by up to 20% and 5%, respectively. This determines the V_s and σ of each layer
5	V_p of each layer can be calculated by $V_p = \frac{V_s}{\sqrt{(1-2\sigma)/(2-2\sigma)}}$. This determines the V_p of each layer
6	Randomly initialize a density ρ (1.818–1.854), and generate a 1.01–1.03 times random increment ρ model. This determines the ρ of each layer
7	Use the generated random layered model to simulate shot gathers by a finite-difference method for modeling Rayleigh (Zeng et al. 2011) and Love waves (Luo et al. 2010); use the generated random layered model to simulate noise recordings (Lawrence et al. 2013); and obtain virtual shot gathers by MAPS (Cheng et al. 2018a, b)

comparative experiment. This threshold energy filtering method can mainly be summarized into three steps: first, to read an RGB (red–green–blue) dispersion image generated from SurfSeis® (developed by Kansas Geological Survey) (Fig. 9a) into an HSV

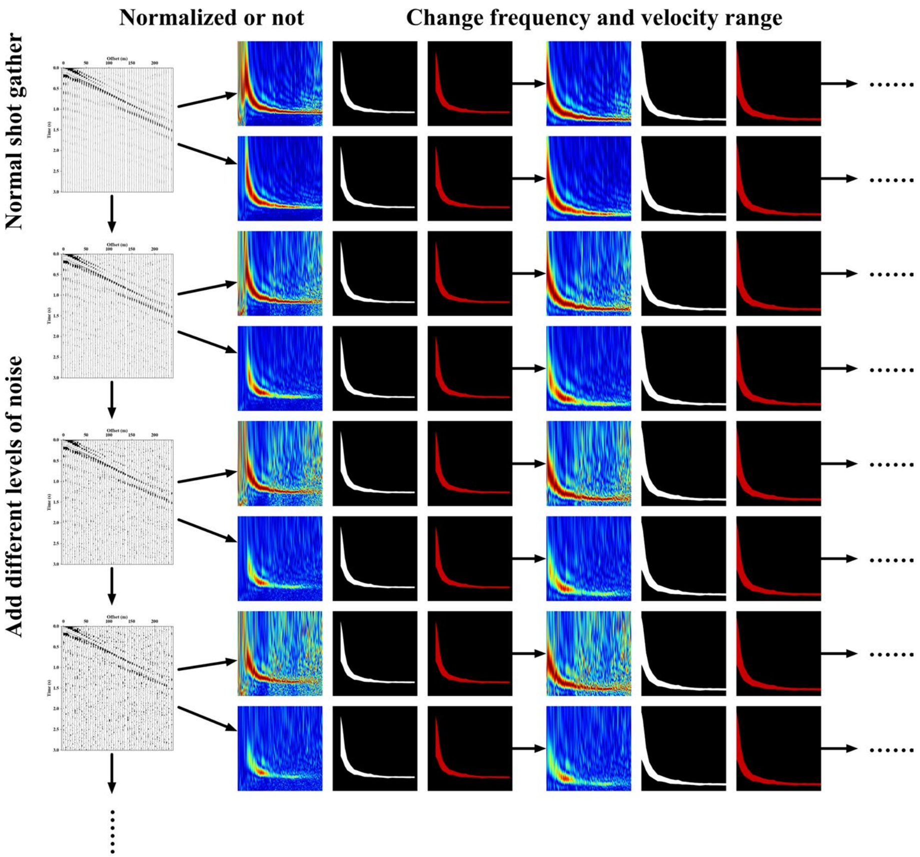

Normalized or not
Change frequency and velocity range
Normal shot gather
Add different levels of noise
Offset (m)
Time (s)
......

Fig. 4 Data augmentation process. One dispersion image, one binary label, and one instance label as one sample. The dispersion images of odd and even rows are normalized and non-normalized, respectively

(hue–saturation–value) representation; second, to select different colors of pixels of the HSV model using different thresholds and mapping them linearly (Fig. 9b) in different ranges (Table 4); third, to find the energy that is higher than 99.9% (red points in Fig. 9d) from local peaks in the 3D dispersion image (Fig. 9c). We extracted the dispersion curves of the same data (Model 2 in Table 3) through our DCNet, and the outputs of the intermediate processes are shown in Fig. 10. Figure 10b and 10c shows the outputs of the segmentation and embedding branches, respectively. We compared the final results (Figs. 9d, 10d) of the extraction by DCNet and threshold energy filtering method with the theoretical dispersion curves (Fig. 11). Calculated by the Formula (4), the mean errors of the fundamental mode dispersion curve extracted by DCNet and threshold energy filtering method are 1.13% and 7.98%, respectively. The mean error of DCNet is just 0.14 times that of the threshold energy filtering method. The mean speed of DCNet is 0.5 s per dispersion image on a personal laptop, 18 times faster than the threshold energy filtering method. Table 5 shows the comparison in detail. The comparison shows that DCNet can extract more accurate and credible dispersion curves in a much shorter time and can separate different modes, which is unable to be done with the threshold energy filtering method.

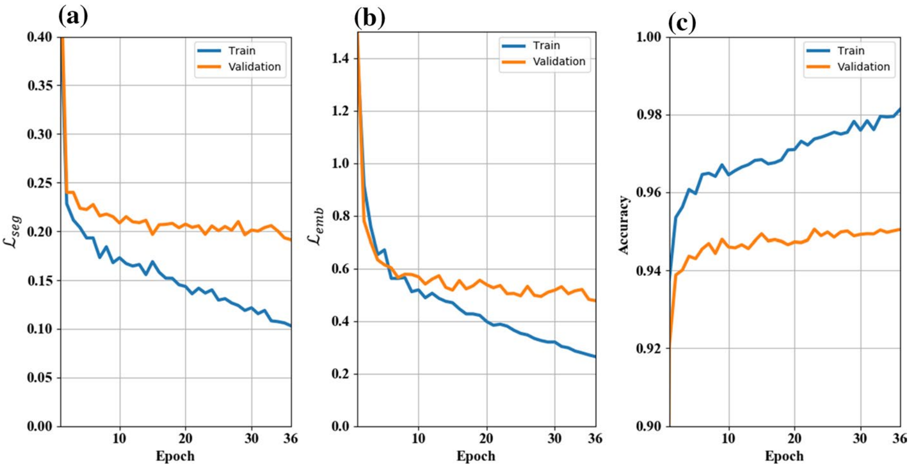


Fig. 5 The training log. **a** Losses of segmentation branch for training (the blue curve) and validation (the orange curve) data sets. **b** Losses of embedding branch for training (the blue curve) and validation (the orange curve) data sets. **c** Mean accuracy rate of DCNet for training (the blue curve) and validation (the orange curve) data sets

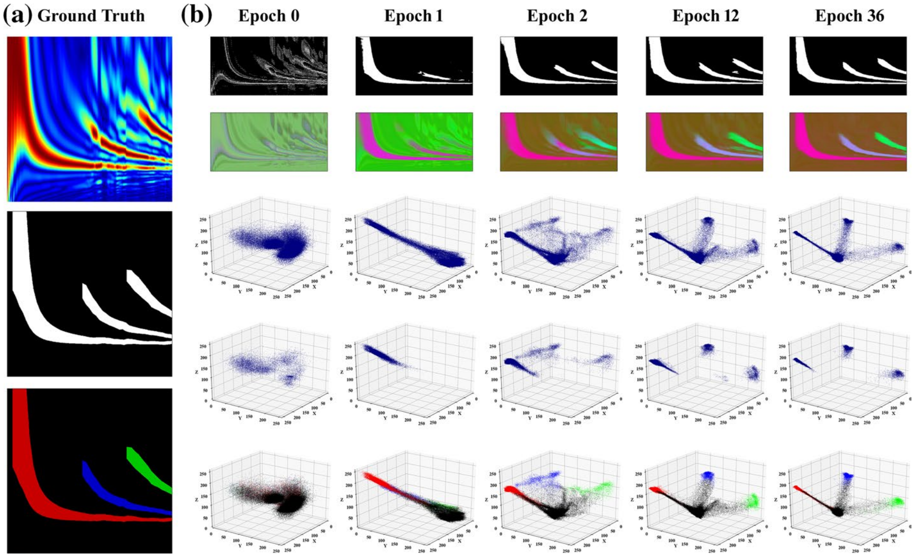


Fig. 6 Convergence of the training process on a dispersion image. **a** The input dispersion image, binary ground truth, and instance ground truth. **b** The outputs of epochs of 0, 1, 2, 12, and 36. The first row represents the outputs of the segmentation branch; the second row represents the outputs of the embedding branch (a 2D representation); the third row represents the outputs of the embedding branch (a 3D representation); the fourth row represents the outputs of the embedding branch (a 3D representation) with outputs of the segmentation branch as mask; and the last row represents the outputs of the embedding branch (a 3D representation) with instance ground truth labels

Table 3 The parameters of two typical simulated models

	Layer number	V_P (m/s)	V_S (m/s)	Density (g/cm^3)	Thickness (m)
Model 1	1	800	200	2.0	10
	2	1200	400	2.0	Infinite
Model 2	1	800	200	2.0	5
	2	1200	400	2.0	5
	3	1600	600	2.0	Infinite

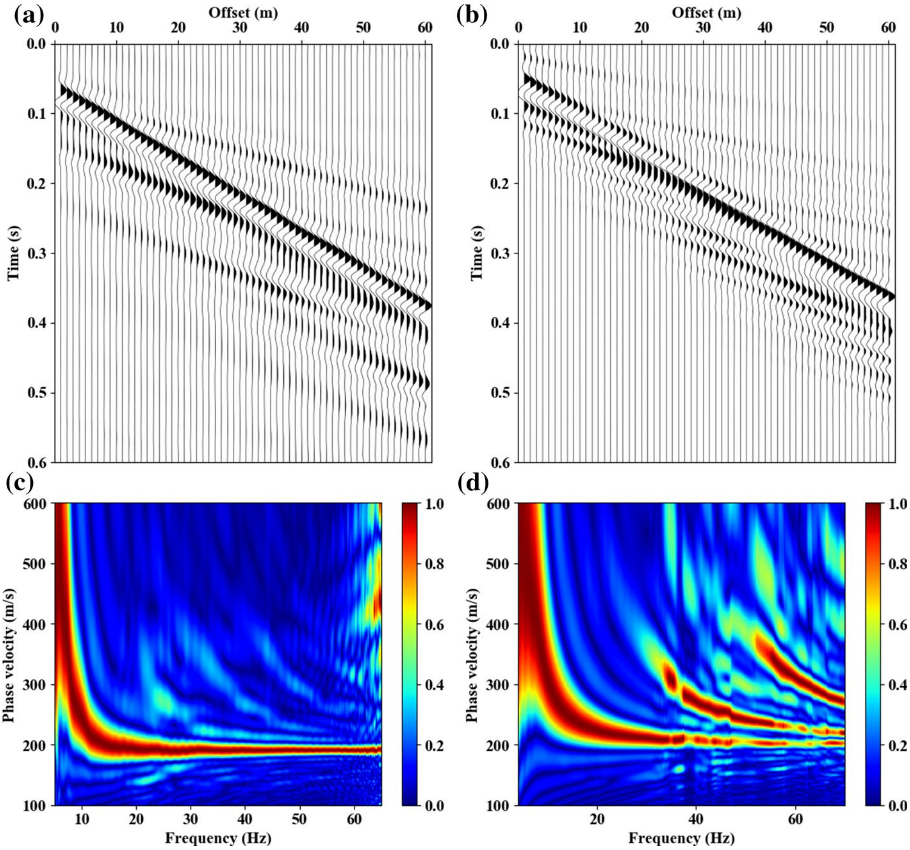


Fig. 7 The shot gathers and dispersion images of the simulated models. **a** The shot gather of Model 1. **b** The shot gather of Model 2. **c** The dispersion image of Model 1 generated using the phase shift method. **d** The dispersion image of Model 2 generated using the phase shift method

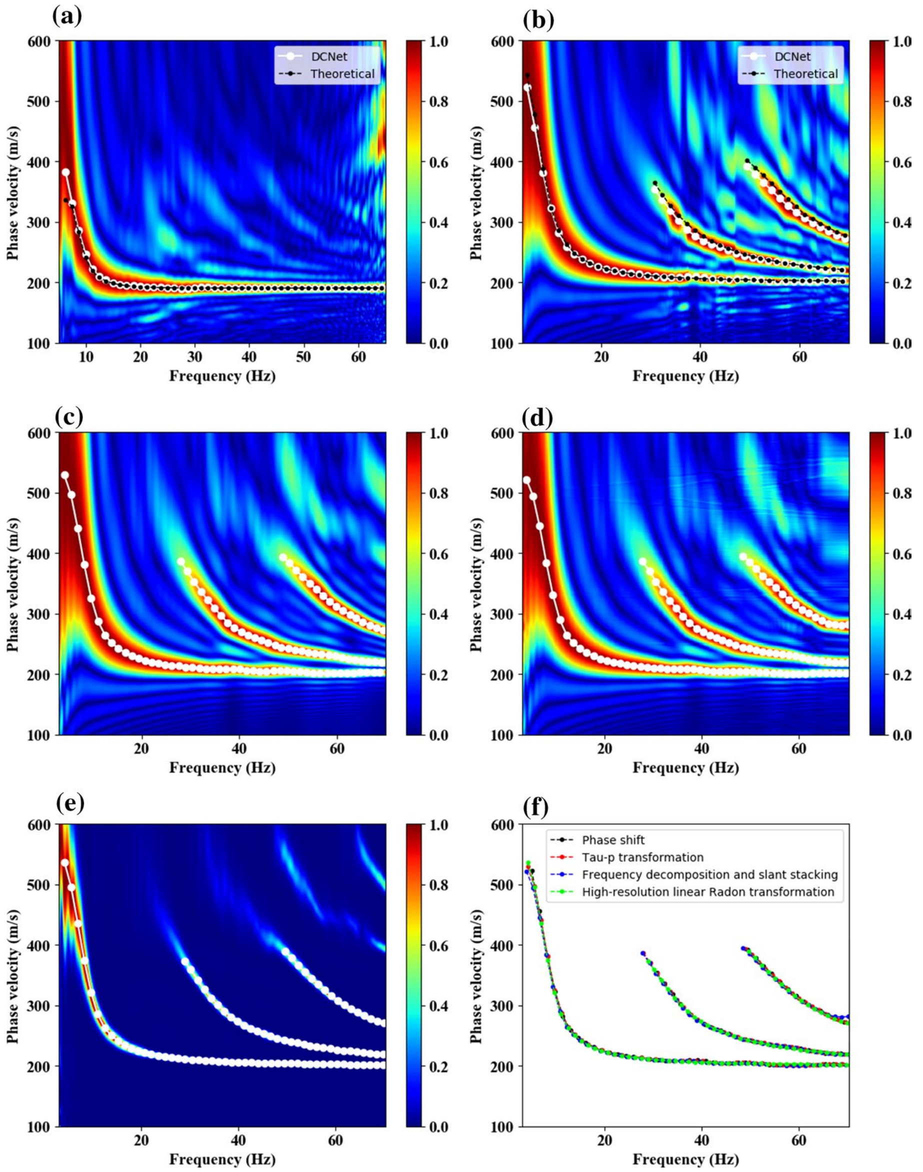


Fig. 8 Comparisons of the dispersion curves extracted by different methods. The white lines with white dots represent the dispersion curves extracted by DCNet; the black dashed lines with smaller dots represent the dispersion curves calculated theoretically. **a** Comparisons of the dispersion curves extracted by DCNet and the theoretically calculated one corresponding to Model 1. **b** Comparisons of the dispersion curves extracted by DCNet and the theoretically calculated one for Model 2. **c** The dispersion image of Model 2 generated using the τ–p transformation. **d** The dispersion image of Model 2 generated using frequency decomposition and slant stacking. **e** The dispersion image of Model 2 generated using HRLRT. **f** Comparisons of the dispersion curves extracted by DCNet using different imaging methods

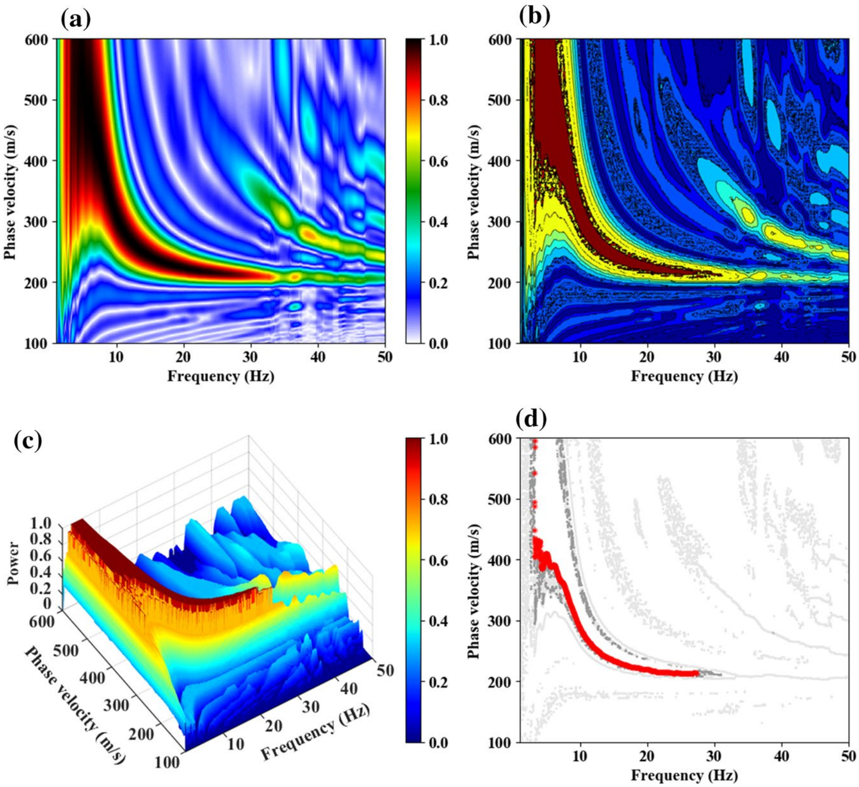


Fig. 9 The process of extracting dispersion curves by a threshold energy filtering method. **a** The dispersion image obtain from SurfSeis® (frequency normalization is not applied). **b** The dispersion image divided into different levels of energy according to the set threshold. **c** The 3D dispersion image representation. **d** Extraction of all possible local peaks from the 3D dispersion image (gray points with different depths representing the extraction of different thresholds) and identification of the fundamental dispersion curve (local highest energy peaks, the red points)

Table 4 Parameters of threshold energy filter method

Color	*H* (Hue)	*S* (Saturation)	*V* (Value)	Range (%)
Blue	(0.566, 0.832)	(0.274, 1.000)	(0.000, 1.000)	0–30
Aqua	(0.103, 0.565)	(0.274, 1.000)	(0.000, 1.000)	30–70
Red	(0.000, 0.102) or (0.897, 1.000)	(0.230, 1.000)	(0.314, 1.000)	70–90
Black	(0.000, 0.102) or (0.897, 1.000)	(0.230, 1.000)	(0.000, 0.314)	90–100

3 Field Data Application

3.1 Fieldwork

Our fieldwork was carried out in the city of Hangzhou, China (Fig. 12). It was a relatively flat area and near two busy roads that are located at the south and west of the field

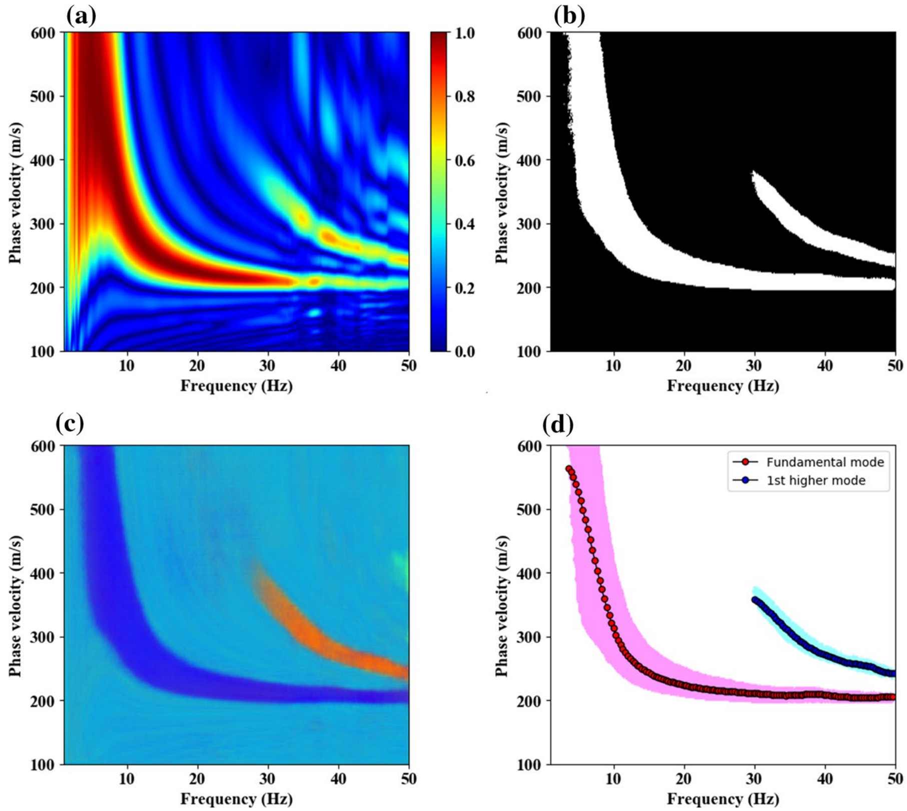


Fig. 10 The process of extracting dispersion curves by DCNet. **a** The same dispersion image as Fig. 9a with a normal color-bar representation (frequency normalization is not applied). **b** Output of the segmentation branch (white areas represent dispersion energy and black areas represent background). **c** Output of the embedding branch (different colors represent different modes). **d** The instance output (light red and blue areas represent energy areas of the fundamental mode and the first higher mode, respectively) and the dispersion curves extracted from the energy of each mode (lines with red and blue dots represent fundamental mode and first higher mode, respectively)

survey area. The test site was covered by silty clay near the tens of meters of near surface. There was a borehole with a depth of 95 m about 50 m from the east of the fieldwork area (the red pentagram highlighted in Fig. 12). The borehole results were used to verify the reliability of the application results.

We designed our field measurements with 14 linear passive survey arrays that contain 13 three-component receivers (the white dots highlighted in Fig. 12) in each array with the 10-m receiver separation (Fig. 13a). The size of the fieldwork area was 130×120 m^2, with a total of 182 three-component receivers with a predominant frequency of 5 Hz placed with a 10-m interval both in north and east directions (black triangles in Fig. 13a). The noise records were recorded with the sampling frequency of 1000 Hz from local time 17:24 on June 15 to 17:19 on June 16 2019.

In the passive seismic data processing of this application, we only utilized the vertical component data to acquire the information of Rayleigh waves. We used the

Fig. 11 Comparison of dispersion curves extracted by DCNet (red lines with red dots) and threshold energy filtering method (the blue dotted line with blue crosses) and theoretical calculation (the black dashed lines with black dots)

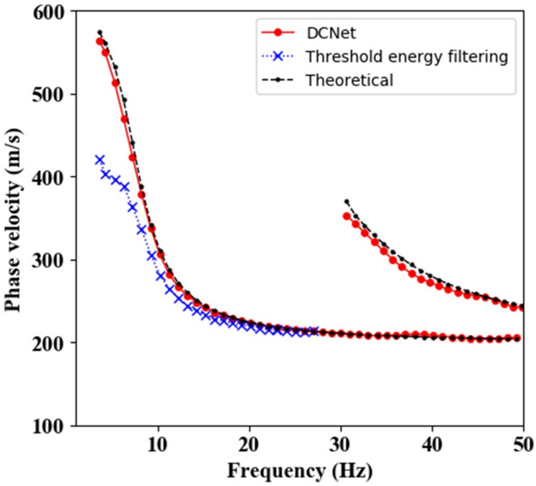


Table 5 Comparisons between DCNet and threshold energy filtering method

Method	DCNet	Threshold energy filtering
Mean error (fundamental)	1.13%	7.98%
Mean speed (s/dispersion image)	0.5	9
Modes number	2	1
N_f (minimum frequency interval is 1 Hz)	68	25
Frequency range (Hz) (fundamental)	3.6–49.3	3.6–28.1

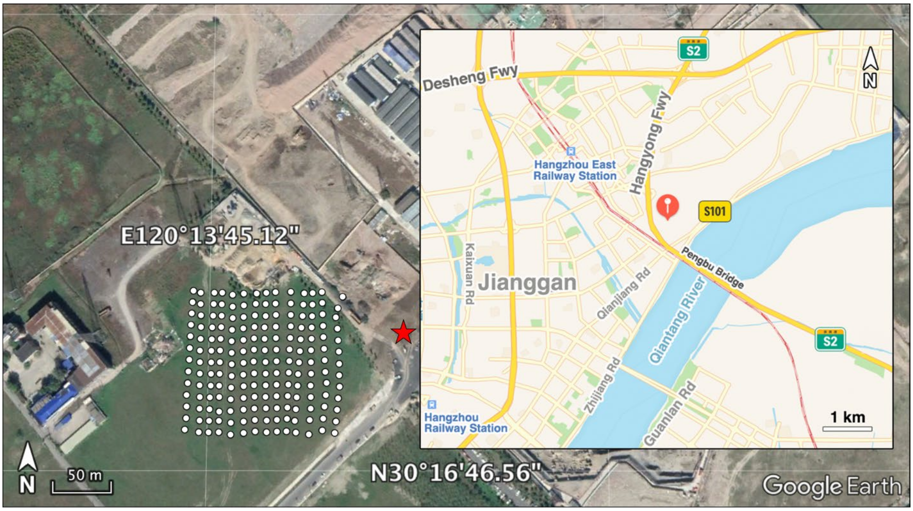


Fig. 12 The location of the fieldwork (produced by Google Earth and Google Maps). The red circle landmark in the map represents the location of the fieldwork area, the white dots in the satellite map represent the true coordinate information of all the three-component receivers, and the red pentagram represents the borehole near the survey line

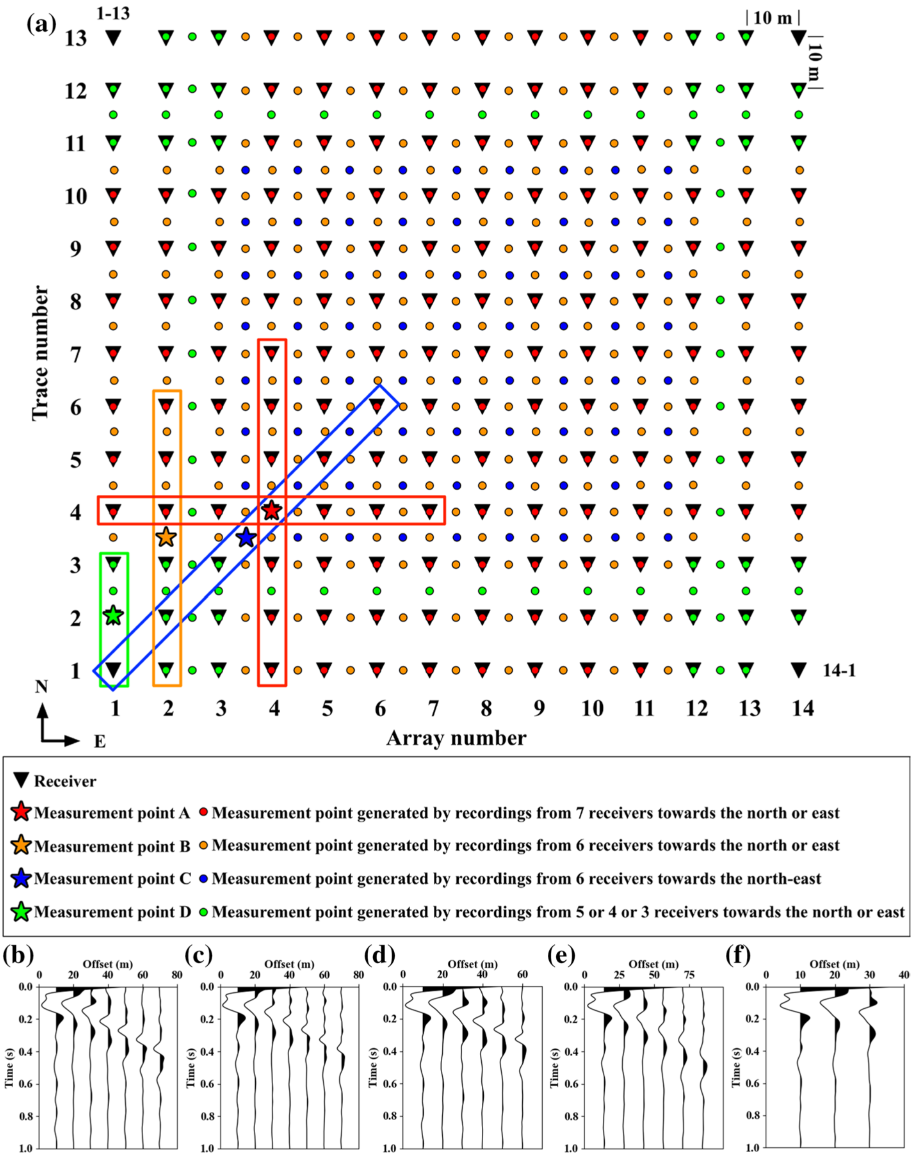


Fig. 13 **a** The field measurements. Black triangles represent the 14 arrays (east–west) × 13 traces (north–south) three-component receivers placed with a 10-m interval both in north and east directions. A receiver is named by "array number–trace number". The colored pentagonal stars represent samples of measurement points. The colored dots represent the measurement points generated by recordings from receivers in different combinations. The colored boxes represent the combinations corresponding to the same colored measurement points. **b**, **c** Virtual shot gathers marked by red boxes for measurement point A; **d** the virtual shot gather marked by the orange box for measurement point B; **e** the virtual shot gather marked by the blue box for measurement point C; and **f** the virtual shot gather marked by the green box for measurement point D

multichannel analysis of passive surface-wave method (MAPS) (Cheng et al. 2016) to process the passive seismic data. After preprocessing, cross-correlation functions between line-arranged traces were used to form the virtual shot gathers according to their internal distance, and then dispersion images were computed by the phase shift method (Park et al. 1998). The 1D S-wave velocity models can be constructed by inverting surface-wave dispersion curves (Xia et al. 1999).

3.2 Extracting Dispersion Curves

Based on the middle-of-receiver-spread assumption (Luo et al. 2009b), implying that the 1D S-wave velocity profile obtained by inversion reflects the medium below the receiver spread, we used recordings from 7 successive receivers to obtain the dispersion curves of the measurement points at the middle of the 7-receiver spread (e.g., the measurement point A generated by recordings from the receivers in the red boxes in Fig. 13a and its virtual shot gathers shown in Fig. 13b, c). The 7-receiver spread is moved 1 receiver position toward the north direction each time, and 7 dispersion curves along array 4 can be obtained. Thus, the dispersion curves of the red measurement points in Fig. 13a can be obtained. In addition, we used recordings from 6 successive receivers to obtain the dispersion curves of the measurement points at the middle of the 6-receiver spread (e.g., the measurement point B generated by recordings from the receivers in the orange box in Fig. 13a and its virtual shot gather shown in Fig. 13d). Thus, the dispersion curves of the orange measurement points in Fig. 13 can be obtained, which made the acquired measurement data denser. We also used recordings from 6 successive receivers in the northeast direction to obtain the dispersion curves of the measurement points in the middle of the 6-receiver spreads (e.g., the measurement point C generated by recordings from the receivers in the blue box in Fig. 13a and its virtual shot gather shown in Fig. 13e). By moving the receiver spread, the dispersion curves of the blue measurement points in Fig. 13a can be obtained. We got a nominal resolution of 5 m × 5 m in the central part of the survey area. We used recordings from 5, 4, or 3 receivers to obtain the measurement data along edges and at corners (e.g., the measurement point D generated by recordings from the receivers in the green box in Fig. 13a and its virtual shot gather shown in Fig. 13f). Therefore, a total of 589 dispersion curves need to be extracted.

We made a list of the receivers required for each measurement point, keeping the same frequency and velocity ranges (Table 6). Then, the dispersion images were calculated in batches. After generating all the dispersion images, all dispersion curves were extracted within 5 min (Fig. 14) with an average speed of only half a second per dispersion image.

Table 6 Examples of the receivers list required for measurement points

Measurement point	Receivers	Spacing interval (m)
A_1	4-1, 4-2, 4-3, 4-4, 4-5, 4-6, 4-7	10
A_2	1-4, 2-4, 3-4, 4-4, 5-4, 6-4, 7-4	10
B	2-1, 2-2, 2-3, 2-4, 2-5, 2-6	10
C	1-1, 2-2, 3-3, 4-4, 5-5, 6-6	14.14
D	1-1, 1-2, 1-3	10

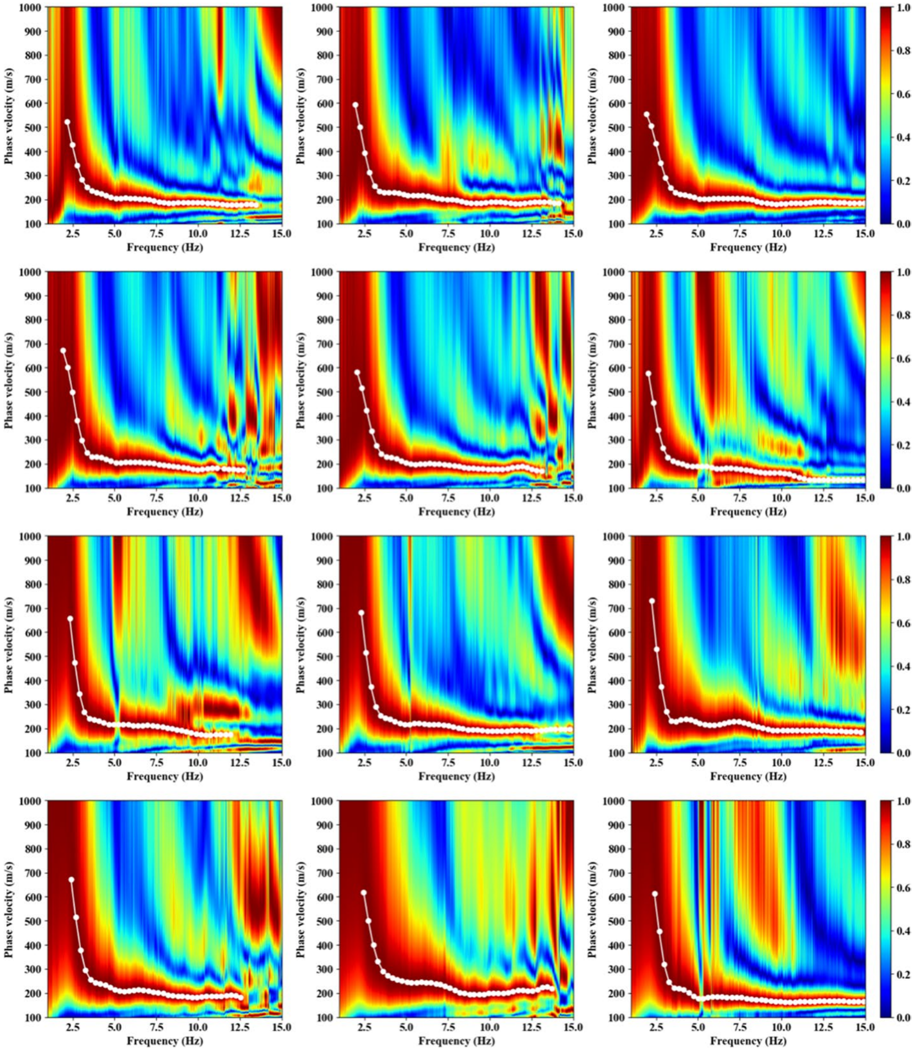


Fig. 14 Examples of the dispersion curves extracted from the field data application. The white lines with white dots represent the dispersion curves extracted by DCNet

This batch processing saved a lot of time on human–machine interaction and did not require the experience of extracting the dispersion curves. For the same measurement location, the dispersion curves obtained by different receiver spreads were averaged to obtain more reliable results (e.g., as Fig. 14 and Table 6 show, measurement point A can be obtained by recordings from the receivers in red boxes toward two directions). Finally, the dispersion curves of 533 measurement points were generated.

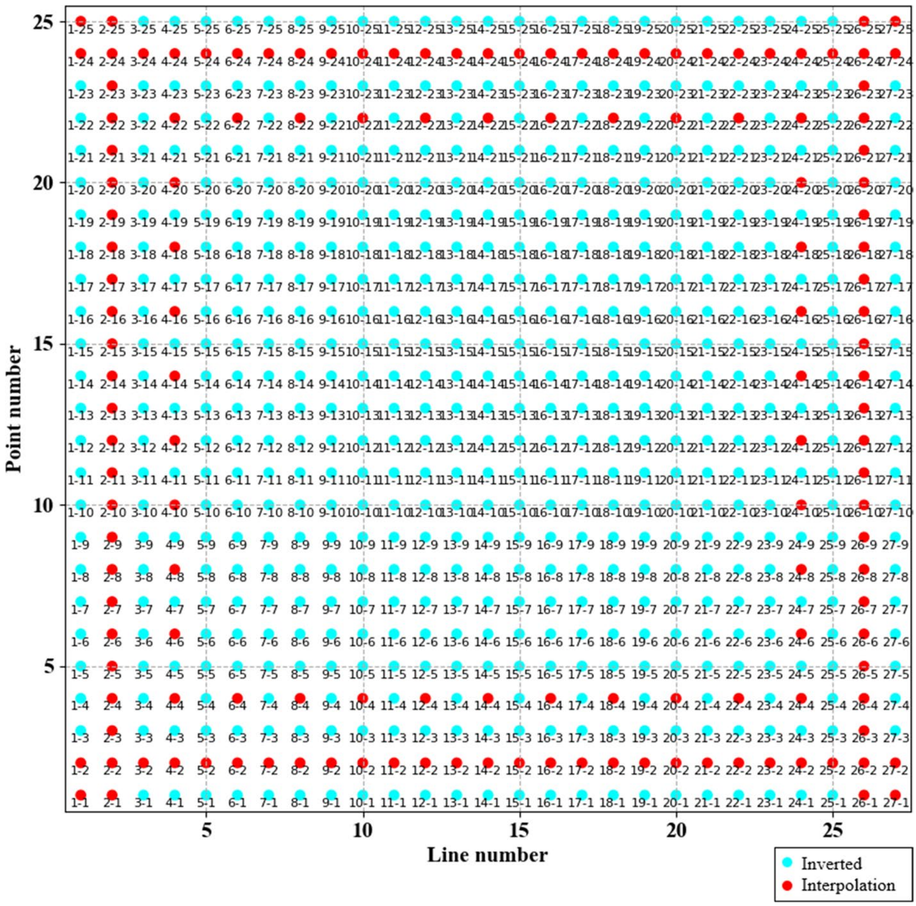


Fig. 15 1D S-wave velocity model assembly. The cyan points represent the inverted 1D S-wave velocity models, and the red points represent the 1D S-wave velocity models generated by linear interpolation. Measurement points are named by "line number–point number"

3.3 3D S-wave Velocity Model

After obtaining a large number of dispersion curves, batch inversion can be carried out with the same initial model. In general, manual intervention was not required; however, occasionally individual points need to be debugged. We obtained a total of 533 1D S-wave velocity profiles, by inverting each extracted dispersion curve independently using a Levenberg–Marquardt algorithm discussed in Xia et al. (1999). Each 1D inverted model like a virtual borehole was placed at its location of the corresponding measurement point (cyan points in Fig. 15). We used the inverted model of the measurement point 27-19 (Fig. 15), which was closest to the borehole, to compare the results with the borehole measurements (Fig. 16a). The RMS error dropped to 6 m/s (Fig. 16c). Our inverted model fits well with the borehole measurements, which shows that the processing results of the field data are reliable. Each virtual borehole at edges and corners (each red point in Fig. 15), which has no extracted dispersion curve for inversion, is obtained via linear interpolation of the

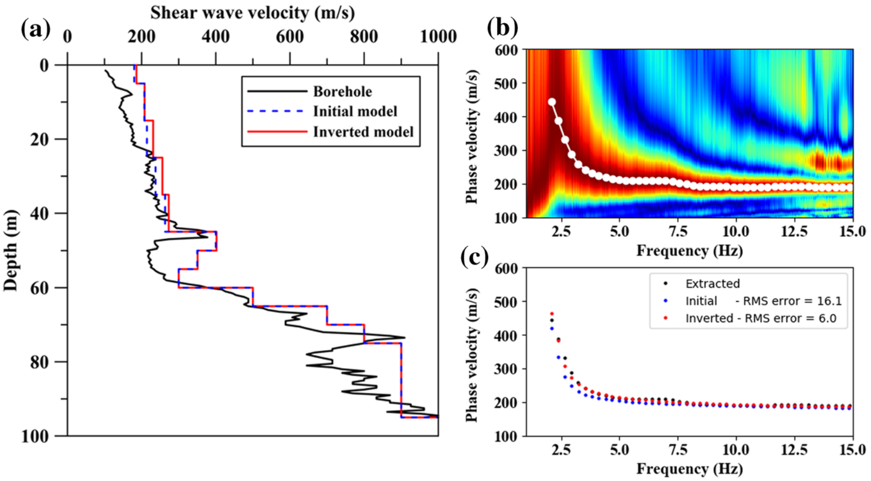


Fig. 16 Comparison with borehole results. **a** The inversion result of the nearest measurement point from the borehole (measurement point 27-19 in Fig. 15). The black line represents the borehole S-wave velocity, the blue dotted line represents the initial S-wave velocity, and the red line represents the inverted S-wave velocity. The S-wave velocity initial model was set according to the borehole data to constrain the inversion results. **b** The dispersion curve extracted by DCNet. **c** The inversion of the dispersion curve. The black dots represent the extracted dispersion curve, the blue dots represent the dispersion curve of initial model with RMS error of 16.1 m/s, and the red dots represent the dispersion curve of the inverted model with RMS error of 6.0 m/s

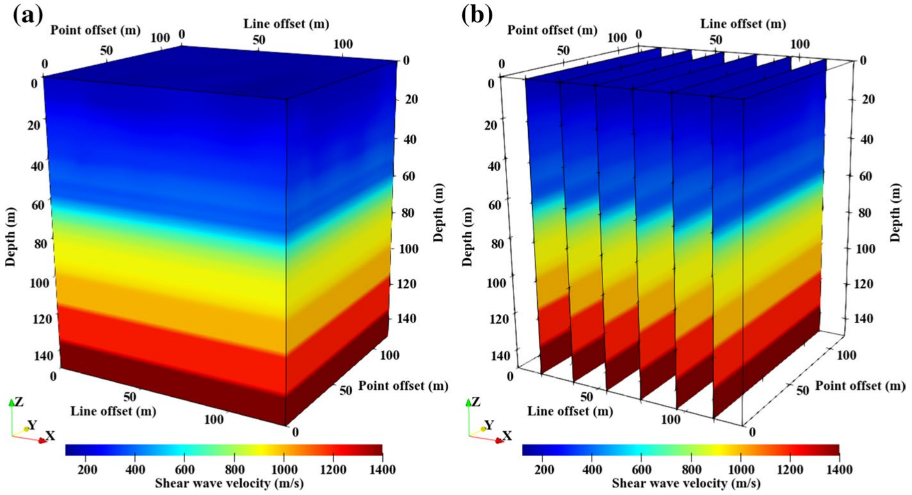


Fig. 17 Final 3D S-wave velocity model of the field data application. **a** Complete 3D S-wave velocity structure via linear interpolation. **b** The vertical sections along the *Y* direction displayed every 20 m from 15 to 115 m

virtual boreholes (cyan points in Fig. 15) from its 8 adjacent positions. We generated a 3D S-wave velocity model (Fig. 17) by assembling all the virtual boreholes. Figure 17b contains vertical sections along the Y direction displayed every 20 m from 15 m to 115 m.

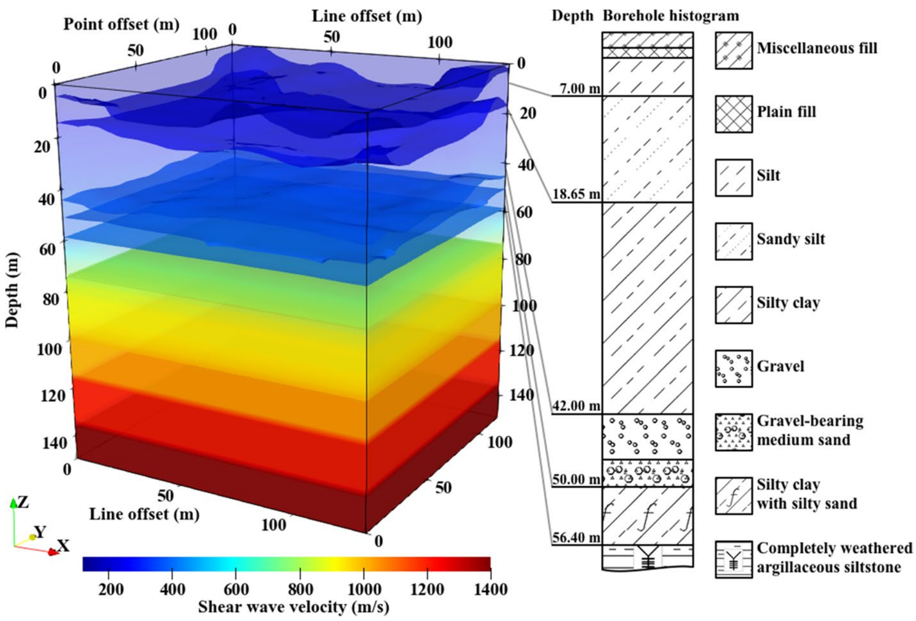


Fig. 18 Isosurfaces of the 3D S-wave velocity model (left) compared with the borehole histogram (right)

It shows that the lateral variation of velocity is small and there is a high-velocity layer at 40 m and 50 m deep.

The isosurfaces of the 3D S-wave velocity model were compared with the borehole histogram (Fig. 18). The S-wave velocity of the first isosurface near the ground surface is 200 m/s, which represents the bottom S-wave velocity interface of silt. The second isosurface is the interface between sandy silt and silty clay. Its S-wave velocity is 250 m/s. The S-wave velocities of third and fourth isosurfaces are 370 m/s, which represent the upper and lower S-wave velocity interfaces of gravel layer. The S-wave velocity of the fifth isosurface is 400 m/s, and then the velocities below the fifth isosurface increase dramatically. It demonstrated that there is a distinct bedrock interface compared to the 56 m depth of borehole histogram. These S-wave velocity isosurfaces are fitting well with the depths of interfaces in the borehole histogram. In the entire process, we only need to set the parameters; then most of the work is done by a computer to obtain such a fine 3D S-wave velocity model, instead of picking the dispersion curves manually point by point as in the past.

4 Discussion and Conclusions

In fact, besides the extracted dispersion curves that can be used for inversion, different outputs of DCNet can be used for different tasks according to the requirements. The output of segmentation branch is exactly the area of surface-wave energy. It can avoid fitting the useless and fake energy in the phase-velocity spectra inversion (Ryden and Park 2006). It can also be used for surface-wave extraction and suppression (Hu et al. 2016). The instance output is exactly the energy areas of different surface-wave modes, which can be used for mode separation by the HRLRT (Luo et al. 2009a). These outputs are used for identifying surface-wave energy instead of manually selection. With the continuous improvement of surface-wave imaging processing (e.g., Bensen et al. 2007; Groos et al. 2012; Ikeda et al. 2013; Shen et al. 2015; Cheng et al. 2018a, b; Zhou et al. 2018; Pang et al. 2019), the quality of the dispersion images is increasing. DCNet can be continuously trained and its training set is expanding all the time. The identification of dispersion energy is becoming easier and easier, and the accuracy of extracted dispersion curves will keep improving. In addition, because of its efficient extraction, this kind of automatic dispersion curves extraction combined with an inversion can produce the real-time profiles, which can offer immediate guidance to fieldwork.

We proposed a deep learning model (DCNet) to rapidly extract numerous multimode surface-wave dispersion curves in the *f*-*v* domain. We also presented a method to generate a large number of labeled surface-wave data for training DCNet model. We compared the theoretical dispersion curves of synthetic data to prove the reliability of our results. By automatically extracting dispersion curves in each window, we obtained a 3D S-wave velocity model by assembling 533 individual inverted 1D S-wave velocity models. The field data application demonstrated the effectiveness and robustness of our method. Data processing automation can improve the efficiency and stability when dealing with such a task with a large amount of data.

Acknowledgements The authors would like to thank associate editor Yu Jeffrey Gu and two anonymous reviewers for their constructive comments and suggestions. This study is supported by the National Natural Science Foundation of China (NSFC) under Grant No. 41774115 and Nanjing Center of China Geological Survey under Grant No. DD20190281. The authors appreciate Binbin Mi, Jingyin Pang, Changjiang Zhou, Hongyu Zhang, and Xinhua Chen for their help in field data collection. The authors also appreciate Xiaojun Chang of Nanjing Center of China Geological Survey for their assistance in field data collection and providing borehole data.

References

Abadi M, Barham P, Chen J, Chen Z, Davis A, Devin M et al (2016) TensorFlow: a system for large-scale machine learning. In: Proceedings of the 12th USENIX symposium on operating systems design and implementation, pp 265–283

Badrinarayanan V, Kendall A, Cipolla R (2017) SegNet: a deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans Pattern Anal Mach Intell 39(12):2481–2495

Bai M, Urtasun R (2017) Deep watershed transform for instance segmentation. arXiv:1802.05591v1 [cs.CV]. https://arxiv.org/abs/1802.05591

Bensen GD, Ritzwoller MH, Barmin MP, Levshin AL, Lin F, Moschetti MP, Shaoiro NM, Yang Y (2007) Processing seismic ambient noise data to obtain reliable broad-band surface wave dispersion measurements. Geophys J Int 169:1239–1260

Boaga J, Vignoli G, Cassiani G (2011) Shear wave profiles from surface wave inversion: the impact of uncertainty on seismic site response analysis. J Geophys Eng 8(2):162–174

Bohlen T, Kugler S, Klein G, Theilen F (2004) 1.5D inversion of lateral variation of Scholte wave dispersion. Geophysics 69(2):330–344

Boiero D, Socco LV (2010) Retrieving lateral variations from surface wave dispersion curves. Geophys Prospect 58(6):977–996

Caruana R (1997) Multitask learning. Mach Learn 28(1):41–75

Chen H, Sun K, Tian Z, Shen C, Huang Y, Yan Y (2020) BlendMask: top-down meets bottom-up for instance segmentation. arXiv:2001.00309v1 [cs.CV]. https://arxiv.org/abs/2001.00309

Cheng F, Xia J, Xu Y, Xu Z, Pan Y (2015) A new passive seismic method based on seismic interferometry and multichannel analysis of surface waves. J Appl Geophys 117:126–135

Cheng F, Xia J, Luo Y, Xu Z, Wang L, Shen C, Liu R, Pan Y, Mi B, Hu Y (2016) Multichannel analysis of passive surface waves based on cross-correlations. Geophysics 81(5):EN57–EN66

Cheng F, Xia J, Xu Z, Mi B (2018a) Frequency-wavenumber (FK)-based data selection in high-frequency passive surface wave survey. Surv Geophys 39(4):661–682

Cheng F, Xia J, Xu Z, Hu Y, Mi B (2018b) Automated data selection in the Tau-*p* domain: application to passive surface wave imaging. Surv Geophys 40(5):1211–1228

Dai J, He K, Sun J (2015) Instance-aware semantic segmentation via multi-task network cascades. arXiv:1512.04412v1 [cs.CV]. https://arxiv.org/abs/1512.04412

Dai T, Hu Y, Ning L, Cheng F, Pang J (2018a) Effects due to aliasing on the surface-wave extraction and suppression in frequency-velocity domain. J Appl Geophys 158:71–81

Dai T, Xia J, Ning L (2018b) Extracting dispersion curves using semantic segmentation of fully convolutional networks. In: Proceeding of 8th international conference on environmental and engineering geophysics, pp 150–155

Daley T, Freifeld B, Ajo-Frankline J, Dou S, Pevzner R, Shulakova V, Kashikar S, Miller D, Götz J, Henninges J, Lüth S (2013) Field testing of fiber-optic distributed acoustic sensing (DAS) for subsurface seismic monitoring. Lead Edge 32(6):699–706

De Brabandere B, Neven D, Van Gool L (2017) Semantic instance segmentation with a discriminative loss function. arXiv:1708.02551v1 [cs.CV]. https://arxiv.org/abs/1708.02551

Ester M, Kriegel HP, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of 2nd international conference on knowledge discovery and data mining (KDD-96)

Evgeniou T, Pontil M (2004) Regularized multi-task learning. In: Proceedings of the 10th ACM SIGKDD international conference on knowledge discovery and data mining, pp 109–117

Foti S, Parolai S, Albarello D, Picozzi M (2011) Application of surface-wave methods for seismic site characterization. Surv Geophys 32:777–825

Groos JC, Bussat S, Ritter JRR (2012) Performance of different processing schemes in seismic noise cross-correlations. Geophys J Int 188:498–512

He K, Gkioxari G, Dollár P, Girshick R (2017) Mask R-CNN. arXiv:1703.06870 [cs.CV]. https://arxiv.org/abs/1703.06870

Howard AG (2013) Some improvements on deep convolutional neural network based image classification. arXiv:1312.5402 [cs.CV]. https://arxiv.org/abs/1312.5402

Hu Y, Wang L, Cheng F, Luo Y, Shen C, Mi B (2016) Ground-roll noise extraction and suppression using high-resolution linear Radon transform. J Appl Geophys 128:8–17

Ikeda T, Tsuji T (2015) Advanced surface-wave analysis for 3D ocean bottom cable data to detect localized heterogeneity in shallow geological formation of a CO_2 storage site. Int J Greenhouse Gas Control 39:107–118

Ikeda T, Tsuji T, Matsuoka T (2013) Window-controlled CMP cross-correlation analysis for surface waves in laterally heterogeneous media. Geophysics 78(6):EN96–EN105

Ivanov J, Miller RD, Lacombe P, Johnson CD, Lane JW (2006) Delineating a shallow fault zone and dipping bedrock strata using multi-channel analysis of surface waves with a land streamer. Geophysics 71(5):A39–A42

Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv:1412.6980 [cs.LG]. https://arxiv.org/abs/1412.6980

Lawrence JF, Denolle M, Seats KJ, Prieto G (2013) A numeric evaluation of attenuation from ambient noise correlation functions. J Geophys Res Solid Earth 188(12):6134–6145

LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444

Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3431–3440

Louie JN (2001) Faster, better: shear-wave velocity to 100 meters depth from refraction microtremor arrays. Bull Seismol Soc Am 91(2):347–364

Luo Y, Xia J, Liu J, Liu Q, Xu S (2007) Joint inversion of high-frequency surface waves with fundamental and higher modes. J Appl Geophys 62(4):375–384

Luo Y, Xia J, Miller RD, Xu Y, Liu J, Liu Q (2008) Rayleigh-wave dispersive energy imaging using a high-resolution linear Radon transform. Pure Apply Geophys 165(5):903–922

Luo Y, Xia J, Miller RD, Xu Y, Liu J, Liu Q (2009a) Rayleigh-wave mode separation by high-resolution linear Radon transform. Geophys J Int 179(1):254–264

Luo Y, Xia J, Liu J, Xu Y, Liu Q (2009b) Research on the middle-of receiver-spread assumption of the MASW method. Soil Dyn Earthq Eng 29:71–79

Luo Y, Xia J, Xu Y, Zeng C, Liu J (2010) Finite-difference modeling and dispersion analysis of high-frequency Love waves for near-surface applications. Pure Appl Geophys 167:1525–1536

Mao B, Han L, Feng Q, Yin Y (2019) Subsurface velocity inversion from deep learning-based data assimilation. J Appl Geophys 167:172–179

McMechan GA, Yedlin MJ (1981) Analysis of dispersive waves by wave field transformation. Geophysics 46:869–874

Mi B, Xia J, Shen C, Wang L, Hu Y, Cheng F (2017) Horizontal resolution of multichannel analysis of surface waves. Geophysics 82(3):EN51–EN66

Mi B, Xia J, Bradford JH, Shen C (2020) Estimating near-surface shear-wave-velocity structures via multichannel analysis of Rayleigh and Love waves: an experiment at the Boise Hydrogeophysical research site. Surv Geophys. https://doi.org/10.1007/s10712-019-09582-4

Neven D, De Brabandere B, Georgoulis S, Proesmans M, Van Gool L (2018) Towards end-to-end lane detection: an instance segmentation approach. arXiv:1802.05591 [cs.CV]. https://arxiv.org/abs/1802.05591

Ning ILC, Sava P (2018) High-resolution multicomponent distributed acoustic sensing. Geophys Prospect 66(6):1111–1122

Okada H (2003) Microtremor survey method. Geophysical Monograph Series, vol 12. Society of Exploration Geophysicists, Tulsa

Ovcharenko O, Kazei V, Kalita M, Peter D, Alkhalifah T (2019) Deep learning for low-frequency extrapolation from multi-offset seismic data. Geophysics 84(6):R1001–R1013

Pan Y, Xia J, Xu Y, Gao L (2016a) Multichannel analysis of Love waves in a 3D seismic acquisition system. Geophysics 81:EN67–EN74

Pan Y, Xia J, Xu Y, Xu Z, Cheng F, Xu H, Gao L (2016b) Delineating shallow S-wave velocity structure using multiple ambient-noise surface-wave methods: an example from western Junggar, China. Bull Seismol Soc Am 106(2):327–336

Pan Y, Schaneng S, Steinweg T, Bohlen T (2018) Estimating S-wave velocities from 3D 9-component shallow seismic data using local Rayleigh-wave dispersion curves—a field study. J Appl Geophys 159:532–539

Pan Y, Gao L, Bohlen T (2019) High-resolution characterization of near-surface structures by surface-wave inversions: from dispersion curve to full waveform. Surv Geophys 40:167–195

Pang J, Cheng F, Shen C, Dai T, Ning L, Zhang K (2019) Automatic passive data selection in time domain for imaging near-surface surface waves. J Appl Geophys 162:108–117

Park CB, Miller RD (2008) Roadside passive multichannel analysis of surface waves (MASW). J Eng Environ Geophys 13(1):1–11

Park CB, Miller M, Xia J (1998) Imaging dispersion curves of surface waves on multi-channel record. In: Society of Exploration and Geophysics (SEG), 68th Annual Meeting, New Orleans, Louisiana, pp 1377–1380

Paszke A, Chaurasia A, Kim S, Culurciello E (2016) ENet: a deep neural network architecture for real-time semantic segmentation. arXiv:1606.02147 [cs.CV]. http://arxiv.org/abs/1606.02147

Perol T, Gharbi M, Denolle M (2018) Convolutional neural network for earthquake detection and location. Sci Adv 4(2):e1700578

Pilz M, Parolai S, Bindi D (2013) Three-dimensional passive imaging of complex seismic fault systems: evidence of surface traces of the Issyk-Ata fault (Kyrgyzstan). Geophys J Int 194:1955–1965

Ren M, Zemel RS (2017) End-to-end instance segmentation with recurrent attention. arXiv:1605.09410 [cs.CV]. http://arxiv.org/abs/1605.09410

Romera-Paredes B, Torr PHS (2016) Recurrent instance segmentation. arXiv:1511.08250 [cs.CV]. http://arxiv.org/abs/1511.08250

Ruder S (2017) An overview of multi-task learning in deep neural networks. arXiv:1706.05098 [cs.CV]. http://arxiv.org/abs/1706.05098

Ryden N, Park CB (2006) Fast simulated annealing inversion of surface waves on pavement using phase-velocity spectra. Geophysics 71(4):R49–R58

Schwab FA, Knopoff L (1972) Fast surface wave and free mode computations. In: Bolt BA (ed) Methods in computational physics. Academic Press, New York, pp 87–180

Shen C (2014) Automatically picking dispersion curves in high-frequency surface-wave method. Master Thesis, China University of Geosciences (Wuhan), Wuhan, Hubei, China

Shen C, Wang A, Wang L, Xu Z, Cheng F (2015) Resolution equivalence of dispersion-imaging methods for noise-free high-frequency surface-wave data. J Appl Geophys 122:167–171

Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556 [Cs]. http://arxiv.org/abs/1409.1556

Socco LV, Boiero D (2008) Improved Monte Carlo inversion of surface wave data. Geophys Prospect 56(3):357–371

Socco LV, Foti S, Boiero D (2010) Surface-wave analysis for building near-surface velocity models—established approaches and new perspectives. Geophysics 75(5):A83–A102

Song X, Zeng X, Thurber C, Wang HF (2019) Imaging shallow structure with active-source surface wave signal recorded by distributed acoustic sensing arrays. Earthq Sci 31:208–214

Taipodia J, Dey A, Gaj S, Baglari D (2020) Quantification of the resolution of dispersion image in active MASW survey and automated extraction of dispersion curve. Comput Geosci 135:104360. https://doi.org/10.1016/j.cageo.2019.104360

Wang F, Chen S (2019) Residual learning of deep convolutional neural network for seismic random noise attenuation. IEEE Geosci Remote Sens Lett 16(8):1314–1318

Wang J, Xiao Z, Liu C, Zhao D, Yao Z (2019) Deep learning for picking seismic arrival times. J Geophys Res Solid Earth. https://doi.org/10.1029/2019JB017536

Wu X, Liang L, Shi Y, Geng Z, Fomel S (2019) Multitask learning for local seismic image processing: fault detection, structure-oriented smoothing with edge-preserving, and seismic normal estimation by using a single convolutional neural network. Geophys J Int 219:2097–2109

Xia J (2014) Estimation of near-surface shear-wave velocities and quality factors using multichannel analysis of surface-wave methods. J Appl Geophys 103:140–151

Xia J, Miller RD, Park CB (1999) Estimation of near-surface shear-wave velocity by inversion of Rayleigh wave. Geophysics 64(3):691–700

Xia J, Miller RD, Park CB, Hunter JA, Harris JB, Ivanov J (2002) Comparing shear-wave velocity profiles from multichannel analysis of surface wave with borehole measurements. Soil Dyn Earthq Eng 22(3):181–190

Xia J, Miller RD, Park CB, Tian G (2003) Inversion of high frequency surface waves with fundamental and higher modes. J Appl Geophys 52(1):45–57

Xia J, Xu Y, Miller RD (2007) Generating image of dispersive energy by frequency decomposition and slant stacking. Pure Apply Geophys 164(5):941–956

Xia J, Xu Y, Luo Y, Miller RD, Cakir R, Zeng C (2012) Advantages of using Multichannel Analysis of Love Waves (MALW) to estimate near-surface shear-wave velocity. Surv Geophys 33:841–860

Yang F, Ma J (2019) Deep-learning inversion: a next-generation seismic velocity model building method. Geophysics 84(4):R583–R599

Yilmaz Ö (1987) Seismic data processing. Society of Exploration Geophysicists, Tulsa, p 526

Yin X, Xu H, Wang L, Hu Y, Shen C, Sun S (2016) Improving horizontal resolution of high-frequency surface-wave methods using travel-time tomography. J Appl Geophys 126:42–51

Zachary ER, Men-Andrin M, Egill H, Thomas HH (2018) Generalized seismic phase detection with deep learning. Bull Seismol Soc Am 108(5A):2894–2901

Zeng C, Xia J, Miller RD, Tsoflias GP (2011) Application of the multiaxial perfectly matched layer to near-surface seismic modeling with Rayleigh waves. Geophysics 76(3):T43–T52

Zhang Z, Alkhalifah T (2019a) Wave-equation Rayleigh-wave dispersion inversion using fundamental and higher modes. Geophysics 84(4):EN57–EN65

Zhang Z, Alkhalifah T (2019b) Regularized elastic full waveform inversion using deep learning. Geophysics 84(5):R741–R751

Zhang G, Wang Z, Chen Y (2018) Deep learning for seismic lithology prediction. Geophys J Int 215:1368–1387

Zhang Z, Alajami M, Alkhalifah T (2020) Wave-equation dispersion spectrum inversion for near-surface characterization using fiber-optics acquisition. Geophys J Int 222:907–918

Zhou C, Xi C, Pang J, Liu Y (2018) Ambient noise data selection based on the asymmetry of cross-correlation functions for near surface applications. J Appl Geophys 159:803–813