scientific reports

OPEN

Gaussian decomposition method for full waveform data of LiDAR base on neural network

Jie Liu, Xinjie Zhang, Jing Lv, Xinyu Li & Libin Du

The full waveform data from airborne LiDAR (Light Detection and Ranging) provides information on the distance of the target. Accurately extracting the ranging information from the full waveform data is crucial for generating point clouds. This paper introduces a method for Gaussian decomposition of full waveform data using a convolutional neural network. The method employs an improved densely connected convolutional neural network and the EM (Expectation Maximization) algorithm to extract information from the data. The method involves two key steps. First, The FWDN network preprocesses the full waveform data to enhance signal quality by reducing noise, and then the improved EM algorithm extracts Gaussian parameters (amplitude, expectation, and full width at half maximum) to obtain ranging information. Based on simulation and measured data, the decomposition success rate of this method is more than 98% with an average range accuracy of less than 1.5 cm compared to other methods. The method has significant potential for application in the field of mapping, 3D modeling.

Airborne LiDAR (Light Detection and Ranging) is an active detection equipment that plays a crucial role in ocean mapping, 3D modeling and military reconnaissance[1–4]. It works by transmitting pulses of laser light and receiving backscattered echoes of the target to measure the distance and other information, which in turn forms the 3D coordinates of the target. LiDAR records the echo intensity of each transmitted pulse, and this data is called full waveform data[5]. The waveform data records the echo intensity magnitude in a time series, which is used to calculate the distance between the LiDAR and the target.

For the existing full waveform algorithms, they can be mainly categorized into three types. They are echo detection, waveform decomposition, and deconvolution[6–8]. Echo detection focuses on extracting the target location directly from the full waveform using local maxima, first order derivative, and thresholding to detect the target location. Waveform decomposition involves decomposing the target echo by combining mathematical function models. The waveform is regarded as the superposition of this function model composition. Commonly used function models include the Gaussian function model, generalized Gaussian function model, and logarithmic Gaussian function. Deconvolution is the process of recovering the target response from the echo waveform using the transmit waveform, provided that the laser transmit waveform and echo waveform are known.

Among these three methods, the echo detection is easy to implement, requires minimal computational effort, and can identify obvious target points. However, it may result in false or missed detections for weak echo targets or echoes mixed with noise. Deconvolution method is capable of obtaining high resolution target reflections, but it is sensitive to noise and clutter, requires data with a good signal-to-noise ratio, and also takes more time than other methods due to the large amount of computation. In contrast, waveform decomposition can better handle situations involving multiple target echoes. By fitting a model to the waveform, the impact of noise on the target echo positions can be reduced, allowing for the extraction of more precise target location information.

The waveform decomposition method typically involves three steps: selecting an appropriate function model to decompose the full waveform data, commonly utilizing the Gaussian function as the base function for decomposition; preprocessing the full waveform, typically through methods such as Gaussian filtering[9], and wavelet decomposition filtering[10], and determining the number of Gaussian components and initial parameters, followed by using an optimization algorithm to fit the preprocessed full waveform data. The Gaussian function is commonly used for waveform decomposition. This is because LiDAR pulses are typically approximated as Gaussian shapes, and when they interact with an object, the resulting echo signals are also approximated as Gaussian shapes[11]. The parameters of the Gaussian function, such as the expectation value and full width at half maximum, correspond to the position and width of the echo signal, making it easy to interpret and analyze.

College of Ocean Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China. email: dulibin@sdust.edu.cn

Hoften[12] et al. were the first to propose decomposing full waveform data with a Gaussian model using the least squares method (LSM) and the Levenberg–Marquardt (LM) algorithm[13] to obtain the Gaussian parameters of the decomposed waveforms. Chauve[14] et al. proposed an augmented Gaussian model to decompose the waveforms, which included the use of a generalized Gaussian function focusing on the extraction of geometric information. Li[15] et al. employed a variable-threshold empirical mode decomposition (EMD) filtering method and an LM optimization algorithm to decompose the original full waveform echo into several independent Gaussian components. Meanwhile, Cheng[16] et al. utilized adaptive noise threshold estimation to eliminate both background and random noise before decomposing the waveform using an improved EM waveform. With the explosive growth of deep learning in recent years, the application of deep learning to full waveform data has become more and more widespread[17–19].

This study proposes a data preprocessing method called One-Dimensional Full Waveform Dense Network (FWDN) that uses convolutional neural networks to learn full waveform data features. FWDN employs one-dimensional convolution and dense connectivity, and the joint loss function optimizes both amplitude and waveform information. An improved EM algorithm is then used to realize Gaussian decomposition. In addition, a complete waveform dataset is created using the collected data to train and validate the network.

Methods

The method for Gaussian decomposition of full waveform data using a convolutional neural network can be divided into two steps. The first step involves preprocessing the raw full waveform data by inputting it into FWDN that performs noise reduction on the data. The next step involves decomposing the full waveform data using a Gaussian model. An improved EM algorithm is used to decompose the Gaussian parameters of the network-processed full waveform data, resulting in the decomposed Gaussian parameters that include amplitude, expectation and full width at half maximum (FWHM). The processing flowchart of the method is shown in Fig. 1.

Gaussian function model

The full waveform data refers to the time-dependent signal received by LiDAR after the laser is reflected by the target. The Gaussian function is a commonly used model for waveform decomposition[20]. The LiDAR emits a laser pulse waveform that approximates a Gaussian waveform. To decompose the full waveform data, the Gaussian function is selected as the base model. This full waveform data is considered to be generated by the superposition of multiple Gaussian waveforms. Therefore, the full waveform data can be expressed in Eq. (1) as:

$$f\left(t\right)=\sum\nolimits_{i=0}^{N}A_{i}e^{\frac{-4\cdot\ln2\cdot\left(t-\mu_{i}\right)^{2}}{\left(\varepsilon_{i}\right)^{2}}}+\eta\left(t\right) \tag{1}$$

where N is the number of components of the Gaussian function, A_i、 μ_i and ε_i are the amplitude, expectation and FWHM of the i th Gaussian component function, $\eta\left(t\right)$ is the noise in the full waveform data, and $f\left(t\right)$ is

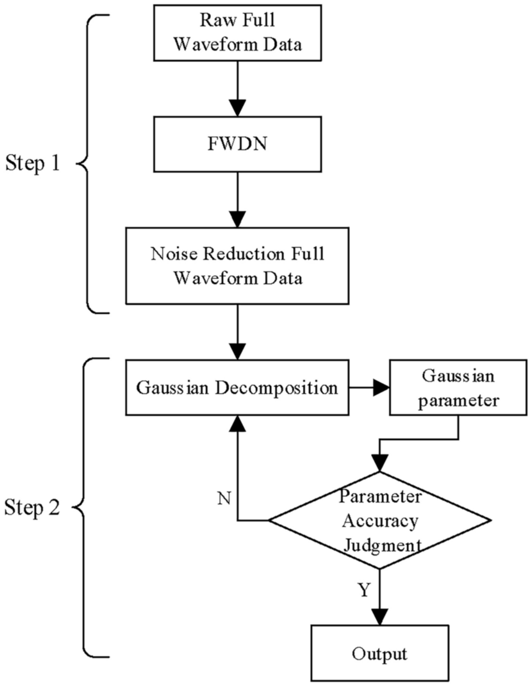

Fig. 1. Flowchart of the method.

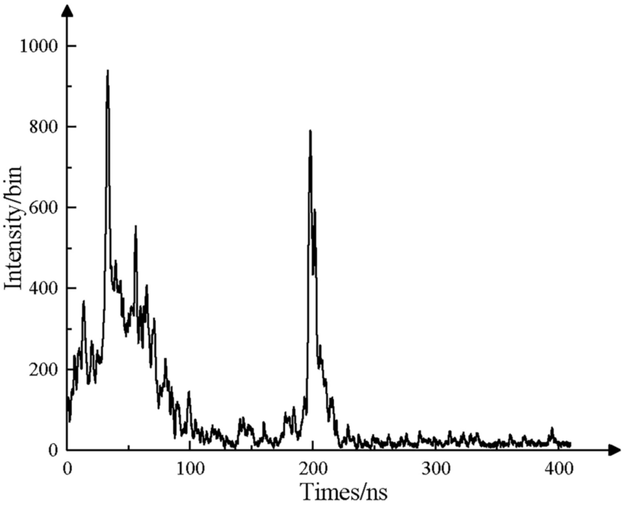

Fig. 2. Raw full waveform signal.

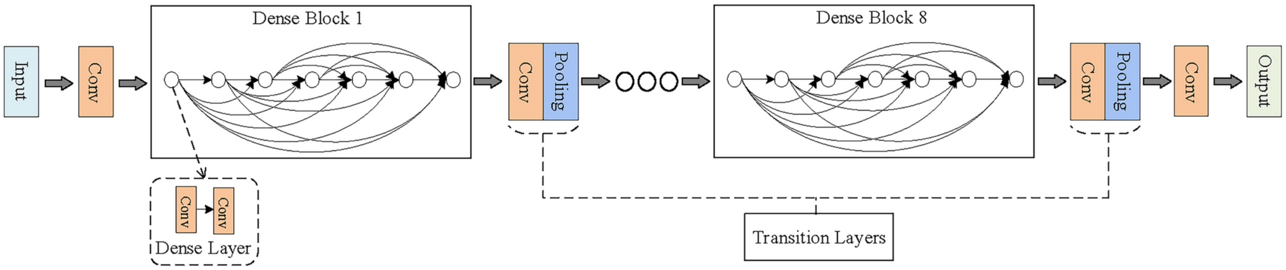

Fig. 3. FWDN Network Architecture.

the full waveform signal. The FWHM are converted to the standard deviation σ in the Gaussian function in Eq. (2) as follows:

$$\varepsilon = 2\sigma\sqrt{2ln2} \tag{2}$$

Each full waveform data can be represented by the function model of Eq. (1), which facilitates subsequent FWDN processing and extraction of Gaussian decomposition parameters.

FWDN

The raw full waveform signal contains both the echo signal of the target and noise signal, as shown in Fig. 2. This noise signal affects the accuracy of the extracted data[21]. Therefore, preprocessing the full waveform data is necessary to improve the quality of the extracted target echo.

In contrast to traditional full waveform data preprocessing, this study uses a Backpropagation (BP) neural network for waveform preprocessing. The BP neural network is trained via the backpropagation algorithm to adjust the weights and biases within the network using the gradient descent method.

The structure of the network is based on densely connected networks[22], and a densely connected network using a one-dimensional convolutional kernel is designed for preprocessing full waveform data. The network's densely connected approach enables the use of features learned in different layers in subsequent layers, making feature transfer across layers more efficient and mitigating the phenomenon of gradient vanishing. This is because each layer is directly connected to the subsequent layer. The network simplifies model complexity and is easy to train and apply by multiplexing and reducing the number of parameters.

The structure of FWDN consists of four parts: the input layer, dense block, transition layer, and output layer. The input layer normalizes the input full waveform data, the dense block extracts full waveform data features, the transition layer compresses the number of data channels and reduces model complexity, and the output layer produces the processing results. The overall structure of the network is shown in Fig. 3.

The input layer receives the full waveform data in batches and undergoes a one-dimensional convolution. This increases the data dimension, allowing for the learning of deeper features from subsequent dense blocks. The convolution layer can be represented by the composite function $H_\iota(\cdot)$, which includes normalization, activation function, and convolution operations.

Each dense block in the network is composed of seven dense layers. These layers are composed of 2 convolutional kernels of different sizes of one-dimensional convolution and activation functions, The current dense layer is directly connected to all previous dense layers. If the output of the current dense layer is denoted as x_ι, it can be expressed as:

$$x_\iota = H_\iota\left([x_0, x_1, \cdots, x_{\iota-1}]\right) \tag{3}$$

The growth rate k is another important parameter of the dense layer. k controls how much the number of channels grows. If the growth rate of each dense layer is k and the number of channels of data before input is k_0 , the number of channels of data passing through dense layers can be expressed as:

$$k_l = k_0 + k * l \tag{4}$$

Each dense block will increase the number of channels, and a transition layer is introduced between adjacent dense blocks to reduce the number of channels. The transition layer uses 1*1 convolution to reduce the number of channels and compress the data dimension. The network parameter structure is shown in Table 1.

The joint loss function is designed as a loss function using Negative Signal to Noise Ratio (Negative SNR) and Mean Square Error (MSE) composition. Using a joint loss function instead of a single loss function allows for the focus on multiple features in the data. Negative SNR is relatively sensitive to changes in amplitude, and MSE measures the degree of difference between the output value and the true value. By using both as loss functions, the purpose of optimizing two objectives is achieved[23]. The loss function is shown in Eq. (5).

$$f_{Loss} = -\lambda_1 SNR\left(\widehat{y}, y\right) + \lambda_2 MSE\left(\widehat{y}, y\right) \tag{5}$$

where λ is the weight coefficient and $\widehat{y}$ and y are the network output value and ideal value. Equation (6) and (7) represent the formulas for SNR and MSE:

$$SNR = 10log_{10}\frac{\|\widehat{y}\|^2}{\|\widehat{y}-y\|^2} \tag{6}$$

$$MSE = \frac{1}{m}\sum_{i=1}^{m}\left(y - \widehat{y}\right)^2 \tag{7}$$

Improved EM algorithm

The computational iterations of the EM algorithm can be divided into an expectation step (E-Step) and a maximization step (M-Step). E-Step estimates parameters based on known data, and then uses them to calculate the expected value of the likelihood function. M-Step calculates the parameters that maximize the likelihood function. These two steps are carried out alternately until the convergence condition is reached. The EM algorithm can fit each Gaussian parameter and obtain the Gaussian parameters that meet the convergence conditions for the Gaussian function model. The formulas are shown in Eqs. (8)-(11).

$$Q_{ij} = \frac{p_j f_j(x_i)}{\sum_{j=1}^{m} p_j f_j(x_i)} \tag{8}$$

$$p_j = \frac{\sum_{i=1}^{n} Q_{ij}}{n} \tag{9}$$

$$\mu_j = \frac{\sum_{i=1}^{n} Q_{ij}\cdot i}{n\cdot p_j} \tag{10}$$

$$\sigma_j = \sqrt{\frac{\sum_{i=1}^{n} Q_{ij}\left(i-\mu_j\right)^2}{n\cdot p_j}} \tag{11}$$

Layers	Kernel size	Output size	Parameters
Input layer	\	$(1\times 1\times 4096)$	\
Convolution	$(1\times 3, 1)$	$(16\times 1\times 4096)$	96
Dense block 1	$\begin{bmatrix}(1\times 5, 1)\\(1\times 1, 1)\end{bmatrix}\times 7$	$(128\times 1\times 4096)$	373,700
Transition layer	$(1\times 1, 1)$	$(16\times 1\times 4096)$	3120
……			
Dense block 8	$\begin{bmatrix}(1\times 5, 1)\\(1\times 1, 1)\end{bmatrix}\times 7$	$(128\times 1\times 4096)$	373,700
Transition layer	$(1\times 1, 1)$	$(16\times 1\times 4096)$	3120
Convolution	$(1\times 1, 1)$	$(1\times 1\times 4096)$	49
Output layer	\	(1×4096)	\

Table 1. Network parameter structure.

where Q_{ij} is the probability of the i th number in the j th component of the Gaussian curve and m is the number of Gaussian components. The EM algorithm's initial values are obtained by prediction, which can lead to convergence to a local optimal solution instead of the global optimal solution. To address this issue, magnitude constraints such as p_j, μ_j, and σ_j are utilized. The improved EM algorithm formulations are presented in Eqs. (12)-(15).

$$Q_{ij} = \frac{p_j f_j(x_i)}{\sum_{j=1}^{m} p_j f_j(x_i)} \tag{12}$$

$$p_j = \frac{\sum_{i=1}^{n} Q_{ij} \cdot N_i}{n \cdot \sum_{i=1}^{n} N_i} \tag{13}$$

$$\mu_j = \frac{\sum_{i=1}^{n} Q_{ij} \cdot N_i \cdot i}{n \cdot p_j \cdot \sum_{i=1}^{n} N_i} \tag{14}$$

$$\sigma_j = \sqrt{\frac{\sum_{i=1}^{n} Q_{ij} \cdot N_i \left(i-\mu_j\right)^2}{n \cdot p_j \cdot \sum_{i=1}^{n} N_i}} \tag{15}$$

Dataset

The impact of the network model is not solely determined by its structure, but also by the dataset used. The full waveform data used in this paper is obtained from the single-wavelength bathymetric LiDAR developed by our research group. The waveform has a time resolution of 0.1 ns, with a sampling length of 4096. Therefore, echo data represent the echo intensity within a time span of 409.6 ns.

Due to the necessity for ideal waveforms in training the network, the raw full waveform data acquired by the LiDAR contains noise and therefore cannot be directly used for network training. Consequently, ideal waveforms need to be generated from the raw full waveform data, and noise is then added to these ideal waveforms to create the full waveform dataset for network training. The waveforms are shown in Fig. 4. GaussPy[24], a python tool for implementing the autonomous Gaussian decomposition, is employed to decompose the raw waveform data, thereby extracting the amplitude, expectation, and FWHM of the waveforms. The aforementioned parameters are used to generate the ideal waveform $y_{id}\left(t\right)$. Subsequently, noise data $\eta\left(t\right)$ collected by the LiDAR is added to the ideal waveform. In this way, a waveform dataset comprising both ideal and noise-containing waveforms is generated. The dataset includes 50,000 ideal waveforms and corresponding noise-containing waveforms, with a distribution ratio of 8:1:1 for the training, validation, and test sets.

Results

Evaluation metrics

In order to better reflect the performance of the FWDN network and compare it with other algorithms, it is essential to define evaluation and comparison metrics. Establish the following four indicators to evaluate the results. The coefficient of determination (R^2) is used to measure the similarity between the output waveform and the ideal waveform, reflecting the network's processing performance. The final decomposition results are evaluated using the separation success rate (S_{fw}), expectation bias (μ_{bias}), and FWHM bias (ε_{bias}) to validate the accuracy of the Gaussian parameters extracted by both FWDN and EM algorithms.

Coefficient of Determination (R^2)

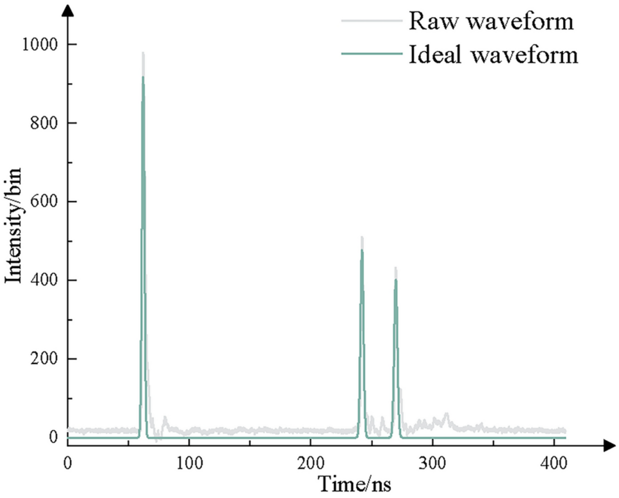

Fig. 4. Raw waveform and ideal waveform.

$$R^2 = 1 - \frac{\sum_{i=1}^{4096}\left(\widehat{y_i}-y_i\right)^2}{\sum_{i=1}^{4096}\left(\overline{y_i}-y_i\right)^2} \tag{16}$$

where, $\overline{y}$ represents the mean of the waveform data. The coefficient of determination (R^2) ranges from 0 to 1, with a higher value indicating a greater similarity between the two waveforms.

Separation Success Rate (S_{fw})

$$S_{fw} = \frac{FW_{succ}}{FW_{sum}} \tag{17}$$

where, the total number of waveforms in the entire dataset is denoted as FW_{sum}. Considering a waveform successfully decomposed when the correct number of Gaussian components is extracted, the count of waveforms with successful decomposition is denoted as FW_{succ}.

Expectation Bias (μ_{bias})

$$\mu_{bias} = \frac{\sum_{i=1}^{M}|\mu_{ci}-\mu_{ai}|}{M} \tag{18}$$

where, μ_c represents the estimated expectation parameter, μ_a is the expectation parameter corresponding in ideal waveform. M is the number of Gaussian parameters in a single waveform. The waveforms used to calculate the expectation bias are those that have been successfully decomposed. Similarly, the following FWHM bias and amplitude bias are used to calculate using successfully decomposed waveforms.

FWHM Bias (ε_{bias})

$$\varepsilon_{bias} = \frac{\sum_{i=1}^{M}|\varepsilon_{ci}-\varepsilon_{ai}|}{M} \tag{19}$$

where, ε_c is the estimated FWHM parameter and ε_a is the corresponding FWHM parameter at the ideal waveform.

Amplitude Bias (A_{bias})

$$A_{bias} = \frac{\sum_{i=1}^{M}\left|A_{ci}^{norm}-A_{ai}^{norm}\right|}{M} \tag{20}$$

where, A_c^{norm} is the estimated amplitude parameter of the normalized waveform and A_a^{norm} is the corresponding amplitude parameter of the ideal waveform after waveform normalization. The normalization process employs a 0–1 normalization approach.

Evaluation of FWDN result

The FWDN network was trained using the training set, and the training environment was built using Python 3.7 and Pytorch 1.11 on an Intel Xeon Platinum 8255C CPU, Nvidia GeForce RTX 3090, and Ubuntu 20.04. The neural network was trained for 128 epochs, and the model parameters corresponding to the epoch with the minimum loss were saved. The loss curve for the network is depicted in Fig. 5. Due to the presence of a negative SNR in the loss function, its value can become negative.

After completing the training of the network, the effectiveness of the network is evaluated using the test set. Ideal waveform, test waveform, and the network's output waveform are shown in Fig. 6. From the figure, it can be observed that there is a high degree of similarity between the output waveform and the ideal waveform. Specifically, the R^2 values calculated for these waveforms are 0.98954 and 0.99472, respectively. Additionally, the average R^2 value calculated for all waveforms in the test set is 0.99043.

Three different filtering methods, namely one-dimensional Gaussian filtering(GF), wavelet soft thresholding(WST) and Empirical Mode Decomposition (EMD) filtering, are selected for comparison with FWDN. The R^2 averages obtained by processing the waveforms of the test set using these three methods are shown in Table 2 along with the R^2 averages of the FWDN. The average R^2 of three methods is much lower than the average R^2 of FWDN, and the degree of similarity between the processed and ideal waveforms is not high.

The results of the four methods applied to the two waveforms in Fig. 6 are shown in Fig. 7. By comparing the five waveforms in Fig. 7 (a) and (b), the waveforms processed by the GF, WST, and EMD methods all have a large range of temporal deviations at the peaks, while the noise filtering is not thorough enough at the other locations. This temporal deviation and incomplete noise filtering may affect the accuracy of the Gaussian decomposition, resulting in a large error between the extracted distance and the actual distance.

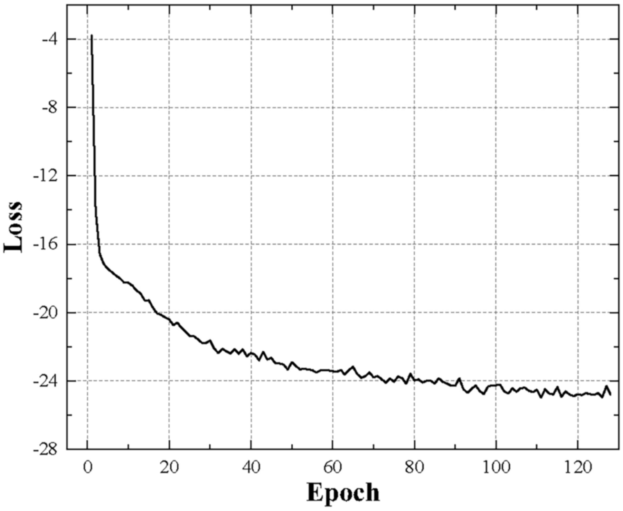

Fig. 5. Training loss curve.

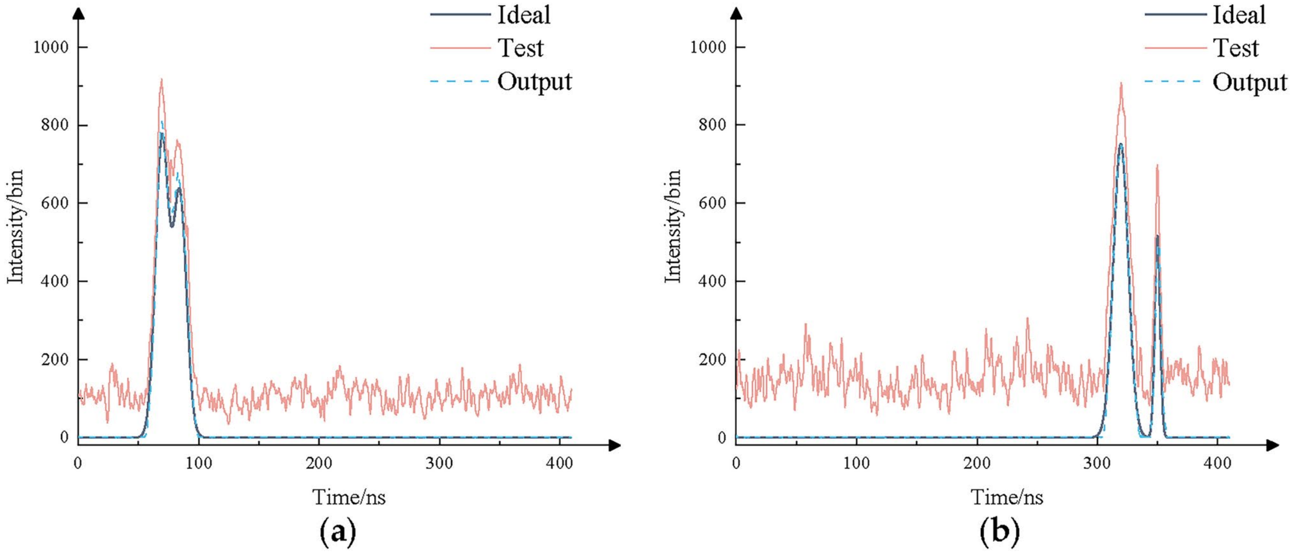

Fig. 6. Comparison of waveforms before and after processing by the FWDN.

Method	GF	WST	EMD	FWDN
R^2	0.43248	0.52536	0.83223	**0.99043**

Table 2. The average R^2 obtained by processing the waveforms in four methods.

Figure 7 shows a comparison of the waveforms processed by the four methods with the ideal waveform. The table in the figure shows the R^2 values of the waveforms processed by these methods. It is evident from the figure that the waveforms processed by GF, WST, and EMD exhibit significant deviations, especially in the peak regions of the target echo. Furthermore, these methods are suboptimal for filtering noise in the flatter regions of the waveform, resulting in insufficient noise suppression. This negatively impacts the accuracy of subsequent Gaussian decomposition of the waveform. Accurate decomposition is crucial for extracting distance features, and these deviations introduce errors between the estimated and actual distances. In contrast, the FWDN method demonstrates superior performance, better handling noise and maintaining closer alignment with the ideal waveform.

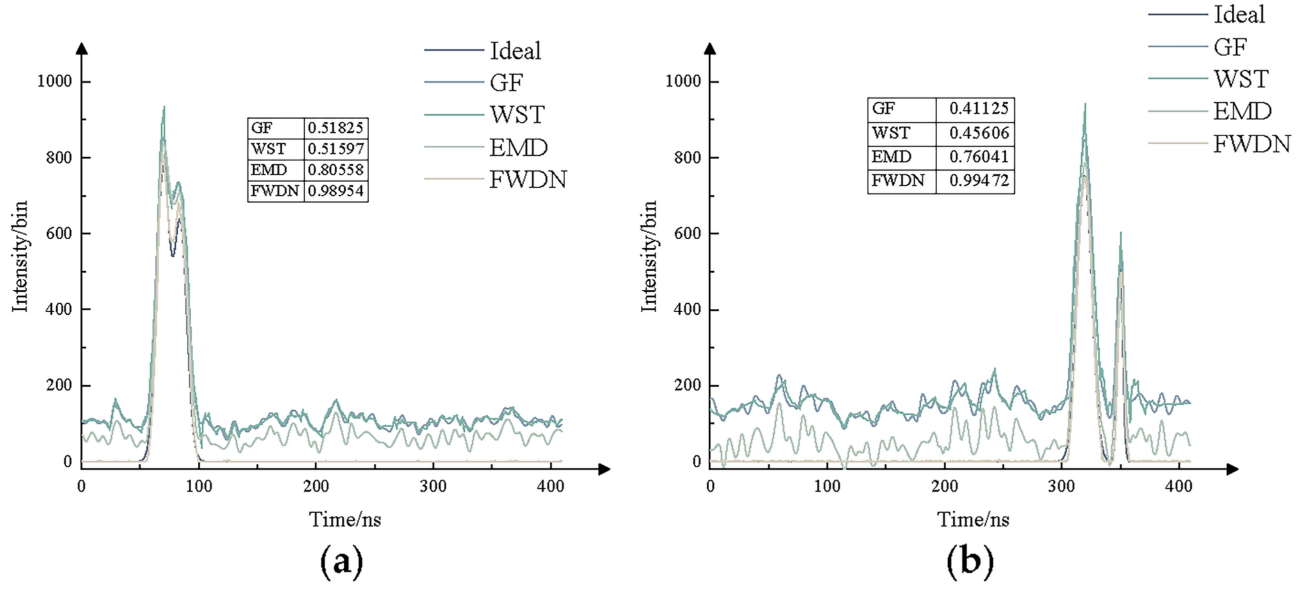

Fig. 7. Comparison of four methods for processing the same waveform.

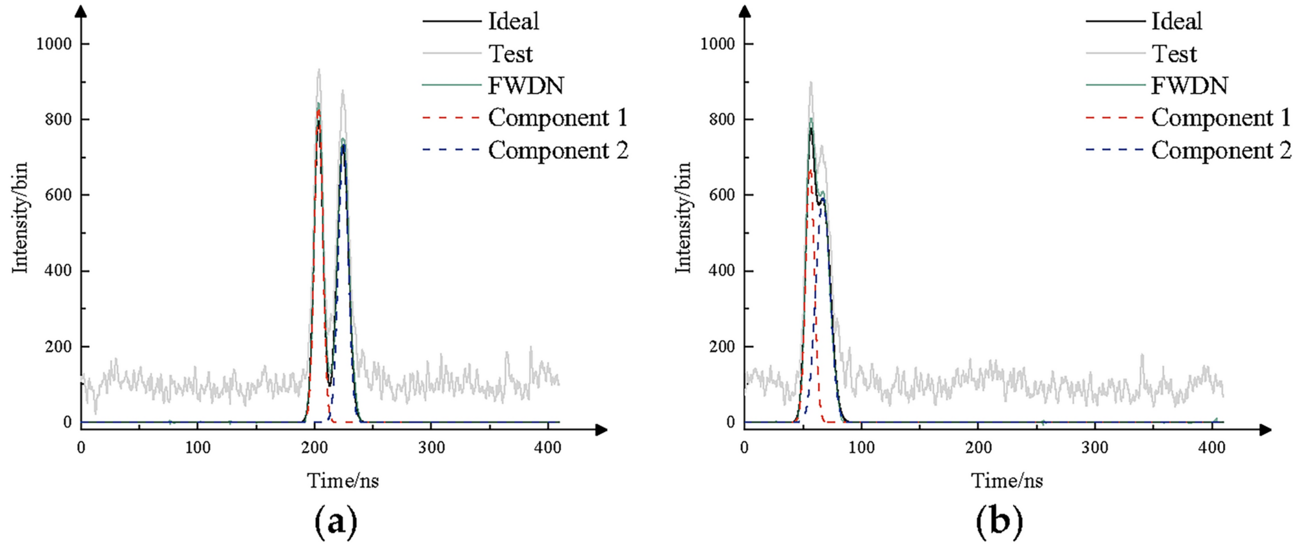

Fig. 8. Gaussian decomposition result.

Evaluation of decomposition result

After network processing, an improved EM algorithm is used to decompose the waveforms, extracting the amplitude, expectation and FWHM parameters of each Gaussian component to describe waveform characteristics. Figure 8 shows two waveforms after processing with FWDN and decomposition using the EM algorithm. The decomposition results demonstrate that the decomposed waveforms closely match the ideal waveforms. Figure 8(a) shows that the Gaussian parameters of the two components can be successfully decomposed when they are separated and do not overlap. Figure 8(b) shows that when the two components are close to each other, their waveforms overlap significantly. The FWDN can restore the waveforms to be almost identical to the ideal waveform, and successfully decompose the Gaussian components of both sets of echoes. The maximum deviation expected in each component is 0.072 ns, which corresponds to a distance of approximately 1.08 cm in the figure.

To verify the precision of the extraction, four evaluation metrics S_{fw}、μ_{bias}、ε_{bias} and A_{bias} mentioned in Section evaluation metrics are used. Use all waveforms of the test set to calculate S_{fw}, and then use the waveforms evaluated by S_{fw} to calculate the average expectation bias ($\overline{\mu}_{bias}$), the average FWHM bias ($\overline{\varepsilon}_{bias}$) and the average amplitude bias($\overline{A}_{bias}$), and their standard deviations are calculated. In addition, the decomposition results of GF, WST and EMD methods are added as a comparison to verify the effectiveness of

Method	S_{fw}	$\overline{\mu}_{bias}$(ns)	$\sigma-\mu_{bias}$(ns)	$\overline{\varepsilon}_{bias}$(ns)	$\sigma-\varepsilon_{bias}$(ns)	$\overline{A}_{bias}$	$\sigma-A_{bias}$
GF *	0.746	1.294	0.598	4.048	2.169	0.066	0.035
WST *	0.783	0.893	0.404	3.101	1.334	0.113	0.036
EMD *	0.925	0.379	0.166	1.973	0.996	0.060	0.020
FWDN *	**0.986**	**0.089**	**0.042**	**1.265**	**0.770**	**0.025**	**0.017**

Table 3. Comparison of evaluation indicators under different methods. *The waveforms processed by these four methods are decomposed using the improved EM algorithm.

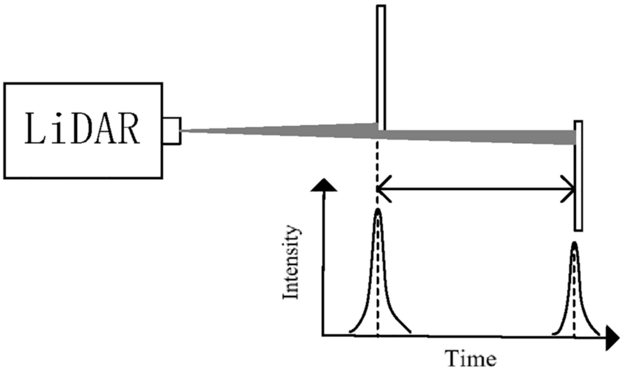

Fig. 9. Schematic diagram of experimental data acquisition.

the method. As evidenced by the results presented in Table 3, the FWDN method demonstrates a commendable degree of efficacy. It is noteworthy that the FWDN method achieves an average expectation of 0.089 ns, which corresponds to a spatial error of approximately 1.33 cm. Furthermore, the method exhibits a low standard deviation of 0.042 ns. Furthermore, the FWDN method demonstrates a notable reduction in FWHM bias, with an average value of 1.265 ns, which is considerably lower than that of the GF (4.048 ns), WST (3.101 ns), and EMD (1.973 ns) methods. With regard to amplitude bias, the FWDN method exhibits a distinct advantage with an average value of 0.025 and a standard deviation of 0.017, surpassing the performance of other methods.

Measured full waveform data decomposition

The full waveform data were obtained using a single-wavelength LiDAR developed by our research group, and the full waveform data were extracted using the method proposed in this paper. The wavelength of the laser utilized in the LiDAR system is 532 nm. The experiment set up two flat plates as targets with a portion of each plate positioned on the laser spot emitted by the LiDAR. By adjusting the distance between the two target plates, the full waveform data containing the two target echoes was acquired, as shown in Fig. 9.

The full waveform data contains the echoes of two flat plates, and the waveform data is decomposed into two Gaussian signals using the Gaussian decomposition method to calculate the distance between the two flat plates. The distance between the two flat plates is measured using a total station, and this value is taken as the true value. Subsequently, the calculated distance is compared with the true value in order to verify the accuracy of the extraction method.

By varying the distance between the two plates, the experiment collected four sets of full waveform data at different distances. The distances measured using a total station were 0.50 m, 6.30 m, 14.40 m and 26.90 m, respectively. Figure 10 shows the full waveform data and its Gaussian decomposition results under these four sets of distances. Specifically, the distances calculated by Gaussian decomposition in Fig. 10 are 0.5135 m, 6.3042 m, 14.3820 m and 26.9195 m, respectively. Table 4 lists the average distance value $\overline{d}$ and standard deviation σ_d . The experimental results indicate that this method demonstrates high accuracy in handling full waveforms at various distances and is capable of accurately decomposing waveform components even under conditions of significant overlap. The error of distance extraction is controlled at the centimeter level with a small standard deviation, indicating the method's.

In order to ascertain the efficacy of the processing method on real collected waveforms, scanning experiments were conducted in an outdoor setting using LiDAR. The resulting collected and processed waveforms are presented in Fig. 11. The figure presents a comparison between the raw waveform, the processed waveform and the result of Gaussian decomposition. This method reduces noise, improves waveform quality, and extracts accurate target distances from the raw waveform.

Conclusion

This study introduces a method for the Gaussian decomposition of LiDAR full waveform data using a convolutional neural network and an improved EM algorithm. The introduction of FWDN effectively filters out the noise, improves the accuracy of Gaussian parameter extraction, achieves centimeter-level ranging accuracy,

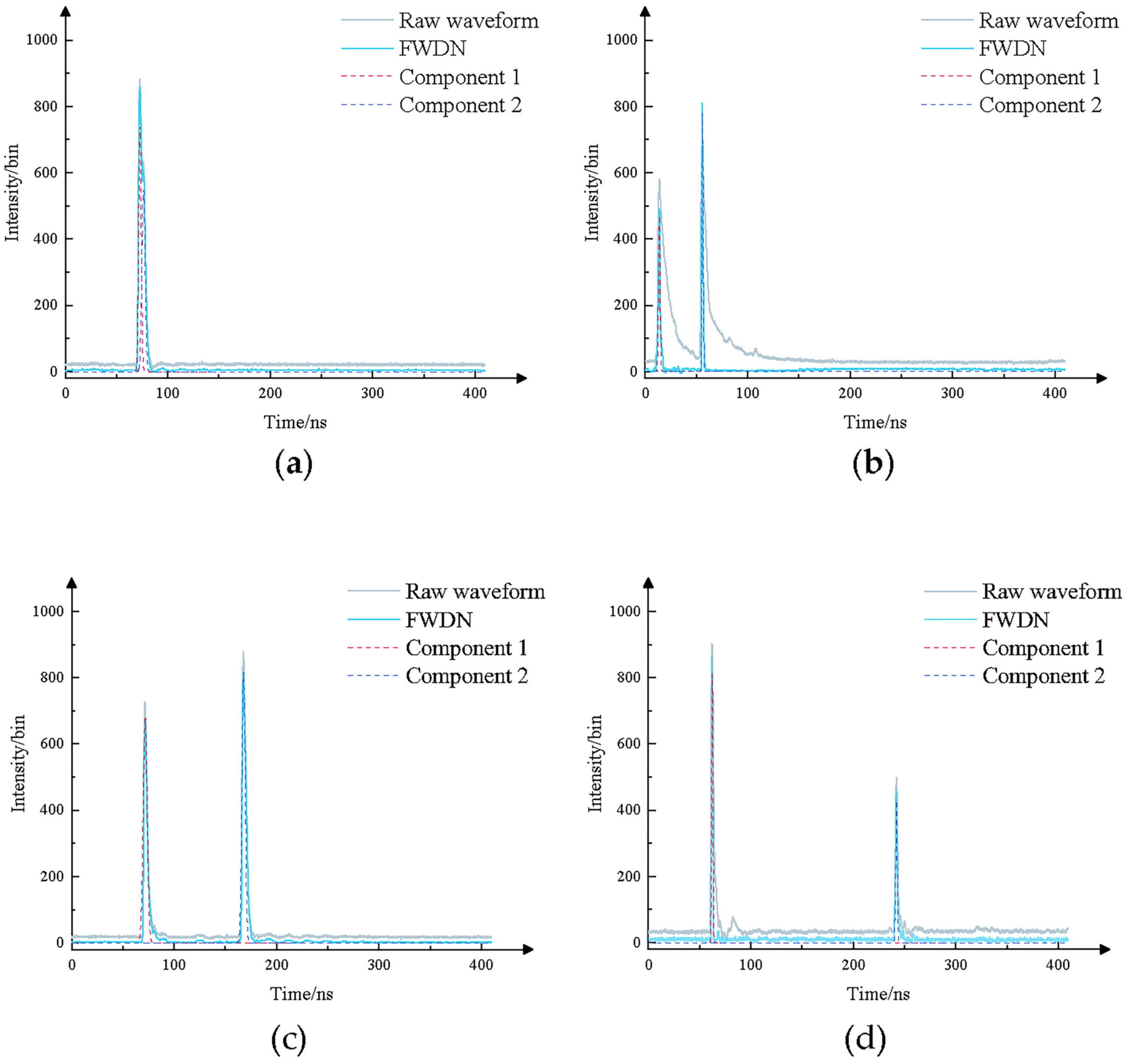

Fig. 10. Acquired waveforms and Gaussian decomposition result.

Distance	Experiment 1 (0.50m)	Experiment 2 (6.30m)	Experiment 3 (14.40m)	Experiment 4 (26.90m)
$\overline{d}$(m)	0.5157	6.2991	14.3895	26.9035
σ_d(m)	0.0179	0.0208	0.0194	0.0225

Table 4. Mean and standard deviation of distances obtained from Gaussian decomposition at different distances.

and enables accurate decomposition in complex multi-echo environments with overlapping waveforms. A comparative analysis of the FWDN method with the GF, WST and EMD methods reveals that it exhibits superior performance across all evaluation metrics, demonstrating better robustness and consistency. The method has significant potential for application in the field of mapping, 3D modeling, and other fields that require high-precision ranging. The further research could concentrate on expanding the data set in order to enhance the adaptability of the network. Furthermore, the implementation of more advanced network architectures can also result in enhanced accuracy and performance.

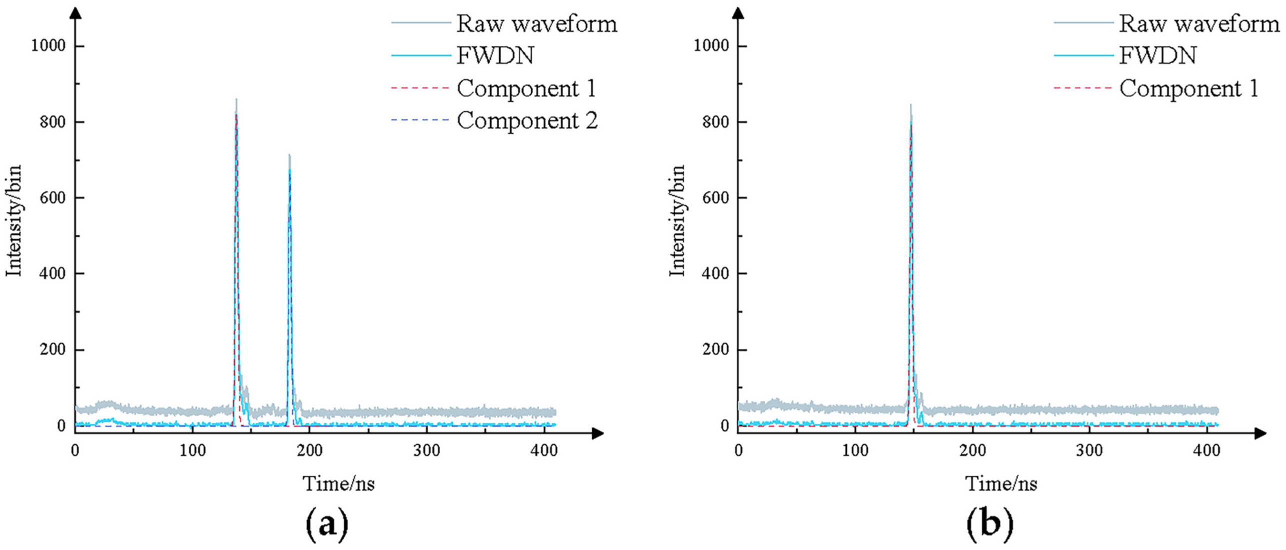

Fig. 11. Waveform and decomposition results.

Data availability

The datasets used in this study are not publicly available, but can be obtained from the corresponding author upon reasonable request.

Received: 24 March 2024; Accepted: 5 December 2024
Published online: 15 February 2025

References

1. Okyay, U., Telling, J., Glennie, C. L. & William, E. D. Airborne LiDAR change detection: An overview of Earth sciences applications. *Earth Sci. Rev.* **198**, 102929 (2019).
2. Sun, S. & Salvaggio, C. Aerial 3D building detection and modeling from airborne LiDAR point clouds. *IEEE J. Select. Top. Appl. Earth Observ. Remote Sens.* **6**(3), 1440–1449 (2013).
3. Gong, W., Shi, S., Chen, B. & Song, S. Development and application of airborne hyperspectral LiDAR imaging technology. *Acta Optica Sinica* **42**(12), 1200002 (2022).
4. Zhao, Y., Li, Y., Shang, Y. & Li, J. Application and development direction of LiDAR. *J. Telem. Track. Command.* **5**, 4–22 (2014).
5. Zhao, X., Xia, H., Zhao, J. & Zhou, F. Adaptive wavelet threshold denoising for bathymetric laser full-waveforms with weak bottom returns. *IEEE Geosci. Remote Sens. Lett.* **19**(1), 1–5 (2022).
6. Zhou, G., Li, W., Zhou, X. & Tan, Y. An innovative echo detection system with STM32 gated and PMT adjustable gain for airborne LiDAR. *Int. J. Remote Sens.* **42**(24), 9187–9211 (2021).
7. Yang, F., Qi, C., Su, D. & Ding, S. An airborne LiDAR bathymetric waveform decomposition method in very shallow water: A case study around Yuanzhi Island in the South China Sea. *Int. J. Appl. Earth Obs. Geoinf.* **109**, 102788 (2022).
8. Song, Y. et al. Comparison of multichannel signal deconvolution algorithms in airborne LiDAR bathymetry based on wavelet transform. *Sci. Rep.* **11**, 16988 (2021).
9. Ma, Y. et al. Noise suppression method for received waveform of satellite laser altimeter based on adaptive filter. *Infrared Laser Eng.* **41**(12), 3263–3268 (2012).
10. Liu, J. et al. Parameter optimization wavelet denoising algorithm for full-waveforms data of laser altimetry satellite. *Chin. J. Lasers* **48**(23), 128–139 (2021).
11. Chauve A., Mallet C., Bretar F. et al. Processing Full-Waveform LiDAR Data: Modelling Raw Signals. ISPRS Workshop on Laser Scanning 2007 and SilviLaser 2007. 102–107 (2007).
12. Hofton, M., Minster, J. & Blair, J. Decomposition of laser altimeter waveforms. *IEEE Trans. Geosci. Remote. Sens.* **38**(4), 1989–1996 (2000).
13. Moré, J. J. The Levenberg-Marquardt algorithm: Implementation and theory. In *Numerical Analysis. Lecture Notes in Mathematics* Vol. 630 (ed. Watson, G. A.) 105–116 (Springer, 1978).
14. Chauve, A. et al. Advanced full-waveform LiDAR data echo detection: Assessing quality of derived terrain and tree height models in an alpine coniferous forest. *Int. J. Remote Sens.* **30**(19), 5211–5228 (2015).
15. Li, H. et al. Full-waveform LiDAR echo decomposition method. *J. Remote Sens.* **23**(1), 89–98 (2019).
16. Cheng S., Zhou M., Yao Q. et al. Full-waveform LiDAR decomposition method using AICC integrated adaptive noise threshold estimation. *Acta Optica Sinica* **42**(12), 279–287 (2022).
17. Hu, S. et al. Classification of sea and land waveforms based on deep learning for airborne laser bathymetry. *Infrared Laser Eng.* **48**(11), 165–172 (2019).
18. Andreas, A., Stewart, B., Wallace, A. M. Deep Learning for LiDAR Waveforms with Multiple Returns. 28th European Signal Processing Conference (EUSIPCO). 1571–1575 (2021).
19. Shinohara, T., Xiu, H., Matsuoka, M. FWNetAE: Spatial representation learning for full waveform data using deep learning. 2019 IEEE International Symposium on Multimedia (ISM). 259–2597 (2019).
20. Li, D., Xu, L., Li, X. et al. A novel full-waveform LiDAR echo decomposition method and simulation verification. 2014 IEEE International Conference on Imaging Systems and Techniques (IST) Proceedings. 184–189 (2014).
21. Parrish, C. E., Jeong, I., Nowak, R. D. & Smith, R. B. Empirical comparison of full-waveform LiDAR algorithms. *Photogrammetr. Eng. Remote Sens.* **77**(8), 825–838 (2011).

22. Huang, G., Liu, Z., Van Der Maaten, L. et al. Densely connected convolutional networks. In: Proc. IEEE conference on computer vision and pattern recognition. 4700–4708 (2017).
23. Sener, O. & Koltun, V. Multi-task learning as multi-objective optimization. *Adv. Neural Inf. Process. Syst.* **31**, 527–538 (2018).
24. Lindner, R. R. et al. Autonomous Gaussian decomposition. *Astron. J.* **149**(4), 138 (2015).

Author contributions

The study concept and design were contributed to by all authors. Professor L. Du provided project support, Professor J. Liu provided guidance on the overall design concept, Mr. X. Zhang wrote the original first draft, Professor J. Lv designed the overall experimental scheme, and Mr. X. Li revised and organized the paper. All authors reviewed the manuscript and approved the final version.

Funding

This work was supported by the Qingdao Marine Science and Technology Collaborative Innovation Center Program of China under Grant (2205CXZX040313).

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to L.D.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.