A Novel Approach to Automatically Digitize Analog Seismograms

Maofa Wang*[1,2], Fengshan Yang[1], Xin Liao[3], Bin Wang[2], Ke Gao[4], Lu Zhang[1], Wenheng Guo[1], Jun Jiang[1], BingChen Yan[1], Yanlin Xu[1], and Quan Wan[1]

Abstract

Before the widespread adoption of the digital seismographs, seismic records were stored in analog form on paper and manually read by analysts. These analog seismograms contained various useful information and were crucial for seismic research. To meet the demands of the modern computational analysis, researchers must digitize historical analog seismograms and extract information. In this article, we present a novel approach to automatically digitize analog seismograms. Initially, Otsu threshold segmentation was applied to the analog seismograms to remove underlying noise and improve their clarity. Subsequently, a novel dynamic distributed seismic waveform onset-point-search algorithm was implemented, which automatically locates the onset point of each seismic waveform baseline in analog seismograms and accurately determines the total number of seismic waveform curves. To address the complexity and diversity of seismic waveforms, we implemented an innovative seismic waveform classification algorithm that can distinguish between complex waveforms and smooth waveforms, and further implemented a new smooth waveform removal method to eliminate interference from smooth waveforms during complex waveform extraction. Then, we used a YOLOv9s-based model to identify time markers within the seismic waveforms for removal. In addition, in the seismic waveform digitization extraction and reconstruction phase, we implemented a novel method for extracting significant seismic waveform features and geometric restoration for peak and trough feature extraction and geometric restoration, as well as vertical feature extraction of seismic waveforms. Finally, we implemented a new waveform sequence integration and time mapping model, which can effectively reconstruct seismic waveform data based on the extracted features and map arrival times to each waveform point. Experiments have verified the significant superiority and stability of the methods implemented in this article for digitizing analog seismograms.

Cite this article as Wang, M., F. Yang, X. Liao, B. Wang, K. Gao, L. Zhang, W. Guo, J. Jiang, B. C. Yan, Y. Xu, and Q. Wan (2024). A Novel Approach to Automatically Digitize Analog Seismograms, *Seismol. Res. Lett.* **XX**, 1–15, doi: 10.1785/0220240220.

Introduction

In the history of seismology, analog seismometers were extensively used to record ground-motion data, primarily dating back to the late nineteenth century. These valuable historical seismic records, preserved on the smoked paper or a film, constitute a rich seismic dataset (Okal, 2015). As time goes by, the storage, management, and analysis of these analog data became increasingly necessary. Especially before the widespread adoption of digital seismometers, these analog records were a valuable resource for analyzing seismic activity (Stein and Wysession, 2003). The U.S. Geological Survey and other relevant agencies have retained a large number of such records, which meticulously document the continuous motion of the ground (Rukstales and Petersen, 2019). Despite advancements in storage technology, many early seismic records are still stored on smoked paper or film, recovering and effectively utilizing this information remains a significant challenge for researchers.

Since the end of the twentieth century, scholars have begun to employ various digitization technologies to convert these analog seismic records into digital formats for better storage and analysis of the data (Ishii *et al.*, 2014; Xu and Xu, 2014; Wang *et al.*, 2016). By utilizing high-resolution scanning technology and image processing software, researchers have been able to extract time-series data from old seismograms using tools such as

1. School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, China, https://orcid.org/0000-0001-6517-4042 (MW); https://orcid.org/0009-0005-2493-2886 (FY); https://orcid.org/0009-0002-4514-3526 (WG); https://orcid.org/0009-0002-0739-6004 (YX); https://orcid.org/0009-0004-5486-125X (QW); 2. Command Center of Natural Resources Comprehensive Survey China Geological Survey, Beijing, China; 3. Institute of Disaster Prevention, Beijing, China, https://orcid.org/0000-0002-5692-5286 (XL); 4. Department of Earth and Space Sciences, Southern University of Science and Technology, Shenzhen, China

*Corresponding author: wangmaofa2008@126.com

DigitSeis and TESEO (TErrain SEismic Observation) developed in the MATLAB (see Data and Resources; Pintore *et al.*, 2005; Bogiatzis and Ishii, 2016). Through the development of multiplatform graphic vectorization and calibration software, like the software used in analyzing the 1928 Parral earthquake in Mexico, researchers can more accurately analyze historical seismic events (Corona-Fernández and Santoyo, 2023).

With the rapid development of deep learning technology, its application in multiple fields has demonstrated significant potential (He *et al.*, 2016; Devlin *et al.*, 2019; Leng *et al.*, 2024). Deep learning has been applied to the processing of analog seismic records, demonstrating potential in handling analog records in seismic control experiments (Wang *et al.*, 2018). Moreover, deep learning techniques such as DevelNet have been proven to effectively detect seismic activity from Develocorder films, offering new possibilities for extracting information from historical seismic data (Wang *et al.*, 2022). Deep learning technologies have also shown their potential in the automatic digitization attempts of strong-motion seismograms at the Japan Meteorological Agency (Furumura *et al.*, 2023).

In summary, the digitization of analog seismic records is a crucial task in seismological research, not only aiding in the preservation of historical seismic information but also providing significant support in understanding seismic processes and reducing earthquake disaster risks (Bungum *et al.*, 2003). To accelerate the digitization of these precious analog seismic record drawings, we propose a novel approach to automatically digitize analog seismograms using deep learning, aimed at achieving rapid and accurate digitization of seismic record waveforms. Specifically, we selected 500 seismograms collected by the Chengde Seismic Station in North China in 1991 as our study subjects. That year, several earthquakes of magnitude 6.5 and above occurred in northern China. These paper seismograms were scanned using professional scanners and converted into raster images, providing a rich data source for our research.

Waveform Structural Analysis and Separation

Threshold segmentation of analog seismograms

In the process of analyzing analog seismograms, we face major challenges including stains, defects, blurred edges, and noise on the seismograms, all of which significantly impact the clarity and accuracy of the analog seismograms. In response to these challenges, we note that these raster format analog seismograms exhibit extreme contrast characteristics in the distribution of gray values, for which most pixels are either very dark (gray value close to 0) or very bright (gray value close to 255), with few intermediate gray levels, indicating high contrast in the seismograms. Based on this observation, we opt to use Otsu's method for analog seismograms threshold segmentation (Otsu, 1979). Otsu's method is an image threshold segmentation technique based on maximizing interclass variance, automatically selecting a threshold to divide the image pixels into two main categories: foreground and background, thereby maximizing the difference between these two categories. By applying the Otsu algorithm, we converted the original grayscale analog seismograms (Fig. 1a) into binary seismograms (Fig. 1b), thus reducing the impact of factors such as stains, defects, blurred edges, and noise, significantly enhancing the accuracy of seismic waveform identification and the overall quality of the analog seismograms.

Dynamic distributed waveform onset search

In our research, we adopted a coordinate system with the origin at the top-left pixel of the seismogram, in which the x axis is positive to the right and the y axis is positive downward.

We implemented a dynamic distributed waveform onset search algorithm, which will be utilized subsequently. This algorithm is used for searching the baseline onset of each waveform and for calculating the number of waveforms in the analog seismograms. Let $\mathbf{I}$ be the matrix representation of the drawing, in which $\mathbf{I}(x,y)$ represents the pixel value at the xth column and yth row. First, The search starts with a fuzzy search, selecting the top-left corner (0,0) of the analog seismograms as the global onset point and scanning downward along the vertical direction y. We identify the transition points for which the pixel color changes from white to black, marked as the upper edge point of waveform i, $y_{\text{upper},i}$. This calculation is shown in equation (1). The scan continues in the same column until the transition from black to white is identified, marked as the lower edge $y_{\text{lower},i}$ of waveform i, as shown in equation (2). Once both the upper and lower edges of a waveform are sequentially and successfully identified, a complete waveform is recognized. Once the scan reaches the bottom of the column, it returns to the global onset point, moves right by n pixel steps, and continues the search in the second column until the set number of columns is searched, completing the fuzzy search. The fuzzy search is illustrated in Figure 2a. The core purpose of fuzzy search is to filter the reference ranges for waveform width and spacing, thereby providing an effective threshold for subsequent precise search. The number of search columns for fuzzy searches can be freely adjusted based on the characteristics of the seismograms to accommodate a variety of seismogram types. The width W_i of waveform i is calculated as shown in equation (3). For vertically adjacent waveforms i and $i+1$, their spacing $D_{i,i+1}$ is calculated as shown in equation (4). All calculated widths W_i and spacings $D_{i,i+1}$ are collected into two datasets W and D, respectively.

$$y_{\text{upper},i} = \min\{y|\mathbf{I}(x,y) = 0 \wedge \mathbf{I}(x,y-1) = 1\}, \tag{1}$$

$$y_{\text{lower},i} = \max\{y|\mathbf{I}(x,y) = 0 \wedge \mathbf{I}(x,y+1) = 1\}, \tag{2}$$

$$W_i = y_{\text{lower},i} - y_{\text{upper},i}, \tag{3}$$

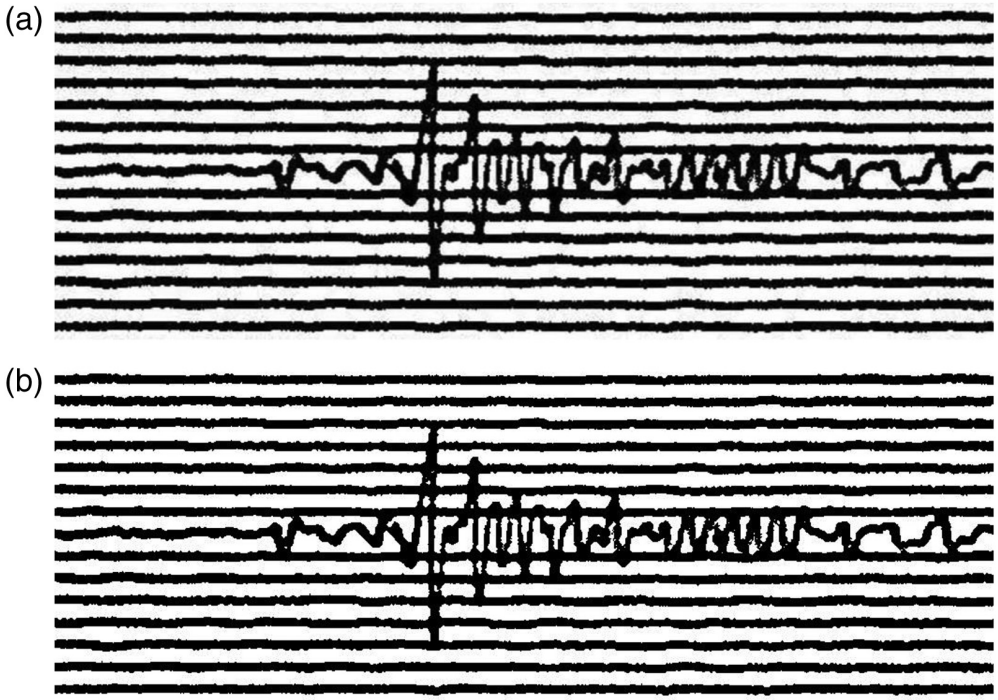

Figure 1. (a) Original grayscale seismogram and (b) binary seismogram.

$$D_{i,i+1} = y_{\text{upper},i+1} - y_{\text{lower},i}. \tag{4}$$

Afterward, outlier processing and statistical analysis are performed on datasets W and D. We calculate the quartiles $Q1$ and $Q3$ and their interquartile range for both W and D (Draper, 2011), as shown in equation (5). Subsequently, based on the statistical theory and widely applied empirical rules, we define the lower bound (LB) and upper bound (UB) for outliers, as calculated in equations (6) and (7). Outliers in W and D that fall below the LB or above the UB are filtered out. Statistical analysis is conducted on the filtered datasets, including the calculation of the mean μ, variance σ^2, and standard deviation σ, as shown in equations (8)–(10).

$$\text{IQR} = Q_3 - Q_1, \tag{5}$$

$$\text{LB} = Q_1 - 1.5 \times \text{IQR}, \tag{6}$$

$$\text{UB} = Q_3 + 1.5 \times \text{IQR}, \tag{7}$$

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i, \tag{8}$$

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2, \tag{9}$$

$$\sigma = \sqrt{\sigma^2}, \tag{10}$$

in which x_i represents an individual observation in the dataset, and n is the total number of observations. Based on the assumption of a normal distribution, data points that fall within the range of $\mu \pm \sigma$ are retained to determine the main concentration trend of the data. From the processed dataset W, the maximum value is selected as $W_{\max}$, and the minimum value as $W_{\min}$. From the processed dataset D, the maximum value is selected as $D_{\max}$, and the minimum value as $D_{\min}$.

Finally, a precise search is conducted, starting from the global origin of the seismogram as the scanning onset point x_{onset} and performing a vertical scan using the same method as in the fuzzy search phase to identify waveforms. If the identified waveform i satisfies equations (11) and (12), then waveform i is valid. If the waveform is valid, the scan continues downward; if not, the onset point is adjusted downward for scanning. For the first identified waveform ($i = 1$), x_{onset} is adjusted to one pixel to the right of the global origin of the seismogram, as calculated in equation (13). For the second and subsequent waveforms ($i \geq 2$), x_{onset} is adjusted to one pixel to the right of the last identified and standard-compliant waveform's lower edge point $x_{\text{lower,last}}$, as calculated in equation (14). Once the scan reaches the bottom of the seismogram, the precise search is completed, as shown in Figure 2b. Both x_{onset} and $x_{\text{lower,last}}$ are x coordinates. The midpoint between the upper and lower edge points of each waveform is taken as the waveform search point, and all waveform search points are moved parallel to the x coordinate of the first waveform search point to establish the baseline onset of the waveforms, as shown in Figure 2c.

$$W_{\min} \leq W_i \leq W_{\max} \quad (i = 1 \ldots n), \tag{11}$$

$$D_{\min} \leq D_{i,i+1} \leq D_{\max} \quad (i = 2 \ldots n), \tag{12}$$

$$x = x_{\text{onset}} + 1, \tag{13}$$

$$x = x_{\text{lower,last}} + 1. \tag{14}$$

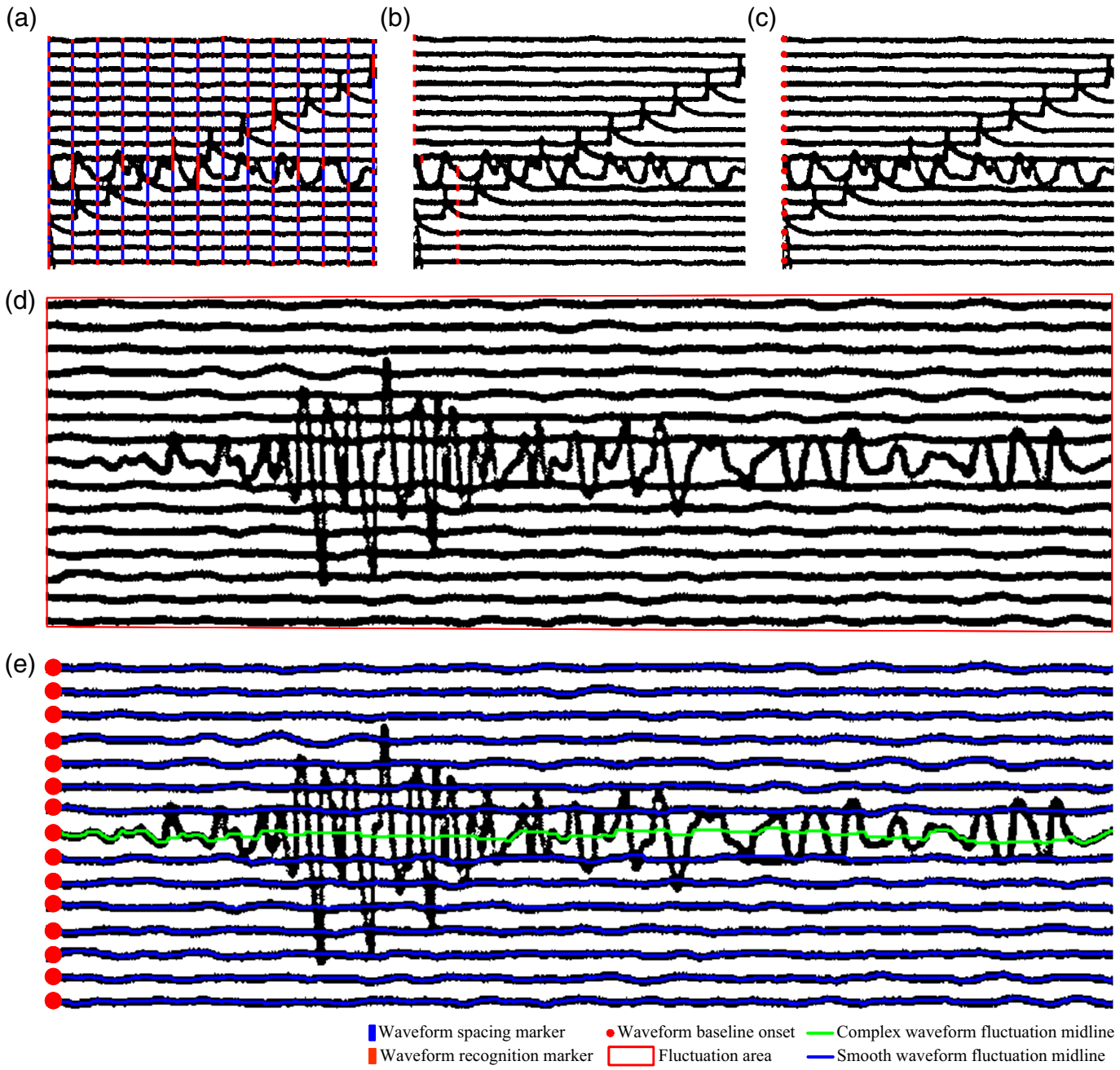

Figure 2. (a) Fuzzy search, (b) precise search, (c) the baseline onset of the waveforms, (d) fluctuation area, and (e) waveform classification. The color version of this figure is available only in the electronic edition.

Waveform classification and removal of smooth waveforms

In this study, seismic waveforms are classified into two categories: smooth waveforms, which exhibit almost no significant fluctuations on the baseline, and complex waveforms, which display noticeable fluctuations on the baseline. To effectively distinguish between these two types of waveforms and mark them for subsequent removal of smooth waveforms, this research employs a two-stage strategy that utilizes the significant differences in waveform characteristics between complex waveforms and smooth waveforms: specifically, the number of white pixels on the baseline of complex waveforms far exceeds that of smooth waveforms. However, since using a straight baseline can cause errors in certain cases, we replace the baseline with a volatility midline when calculating the number of white pixels.

Initially, the dynamic distributed waveform onset search algorithm is used to determine the baseline onset of all waveforms within the manually selected fluctuation area (Fig. 2d). Subsequently, we perform a lateral fluctuation scan from the baseline onset along the waveform to the endpoint (edge of

the fluctuation area) and calculate the number of white pixels on the volatility midline. If a waveform has the highest number of white pixels on its volatility midline, it is classified as a complex waveform; otherwise, it is classified as a smooth waveform (Fig. 2e). The specific steps are as follows.

1. Use the dynamic distributed waveform onset search to scan the fluctuation area to determine the baseline onset for each waveform i, $(x_{\text{onset},i},y_{\text{onset},i})$.
2. For each waveform i within the fluctuation area, select the baseline onset $(x_{\text{onset},i},y_{\text{onset},i})$ as the starting point. Check the pixel value $\mathbf{I}(x_{\text{onset},i},y_{\text{onset},i})$.
   - If $\mathbf{I}(x_{\text{onset},i},y_{\text{onset},i}) = 255$ (white), it is directly considered as the fluctuation midline point.
   - If $\mathbf{I}(x_{\text{onset},i},y_{\text{onset},i}) = 0$ (black), perform a vertical scan with a step size of one pixel along the positive y axis to $(x_{\text{onset},i},y_{\text{onset},i} + 2\Delta y)$ (Δy is the average width of the waveform, obtained through the dynamic distributed waveform onset search algorithm). Check for any color transition points within this area. If a transition point is found, record it as max. If no transition point is detected, use the previous max value. Then, scan along the negative y axis to $(x_{\text{onset},i},y_{\text{onset},i} - 2\Delta y)$ and similarly check for color transition points. If a transition point is found, record it as min. If no transition point is detected, use the previous min value. Calculate the midpoint between max and min as p. Set $(x_{\text{onset},i},p)$ as the fluctuation midline point.
3. Update the coordinates $x_{\text{onset},i} = x_{\text{onset},i} + 1$, $y_{\text{onset},i} = p$.
4. Repeat steps (2) and (3) until the waveform has been completely scanned. Then, move the algorithm to the baseline onset of the next waveform and repeat the entire process until all waveforms in the fluctuation area have been scanned.
5. Check the pixel value $\mathbf{I}(x,y)$ at each fluctuation midline point for each waveform. If $\mathbf{I}(x,y) = 255$ (white), count it toward the total number of white pixels. For each waveform i, the total number of white pixels on its fluctuation midline B_i is calculated using equation (15), in which $[\mathbf{I}(x,y) = 255]$ is an indicator function that checks if the pixel at position (x,y) is white, $[\text{waveform}_i]$ is the set of fluctuation midline points for waveform.
6. After calculating the total number of white pixels on the fluctuation midlines of all waveforms, mark the waveform with the highest B_i value as a complex waveform, whereas the others are marked as smooth waveforms.

In the waveform classification algorithm described earlier, the selected fluctuation area contains only one complex waveform. For cases where the fluctuation area contains multiple complex waveforms, a threshold for white pixel count can be set according to the characteristics of different seismograms to select multiple complex waveforms.

$$B_i = \sum_{(x,y)\in[\text{waveform}_i]} [\mathbf{I}(x,y) = 255]. \quad (15)$$

After classifying the waveforms, we propose a structure-based smooth waveform removal algorithm to obtain complex waveforms without interference from smooth waveforms. The specific steps are as follows.

1. From the fluctuation area, select the baseline onset of the first waveform $(x_{\text{onset1}},y_{\text{onset1}})$.
2. Vertically, fix the x coordinate x_{onset1}, and with a step size of one pixel, first scan in the positive y direction up to $(x_{\text{onset1}},y_{\text{onset1}} + \Delta y)$ (Δy is the average waveform width, obtained from the dynamic distributed waveform onset search algorithm). Check for color transition points within this area, and then scan in the negative y direction up to $(x_{\text{onset1}},y_{\text{onset1}} - \Delta y)$, also checking for color transition points within this area. If color transition points appear in both the upper and lower directions within the area $(y_{\text{onset1}} + \Delta y,y_{\text{onset1}} - \Delta y)$, convert all black pixels in this range to white pixels; otherwise, leave unchanged.
3. Update the x coordinate $x_{\text{onset1}} = x_{\text{onset1}} + 1$, setting the step size to one pixel value.
4. Repeat steps (2) and (3) until the processing of this waveform is completed. Subsequently, the algorithm moves to the next smooth waveform's onset point and repeats the entire process until all smooth waveforms within the fluctuation area have been removed (the algorithm description is shown in Fig. 3).

Smooth waveforms are eliminated using the structure-based smooth waveform removal algorithm, thereby excluding their interference with the subsequent extraction of complex waveforms.

YOLOv9 identifies time markers

In the seismogram analysis, time markers are key reference points indicating the time axis of seismic waveforms. These time markers enable us to perform precise time calculations. However, when time markers overlap with complex waveform regions, the removal of smooth waveforms can leave behind these markers, thus interfering with subsequent waveform extraction. We employ YOLOv9 to identify the time markers, which can then be used for time calculations and removed to prevent interference with waveform extraction.

YOLOv9 is an advanced object detection model chosen for its balance between processing speed and accuracy (Wang, Yeh, and Liao, 2024). The model incorporates programmable gradient information (PGI) technology and a Generalized Efficient Layer Aggregation Network (GELAN) architecture. PGI optimizes gradient transmission by introducing auxiliary reversible branches, ensuring that the key gradient information is not lost in the multilayer network, thus enhancing the model's convergence speed. The GELAN

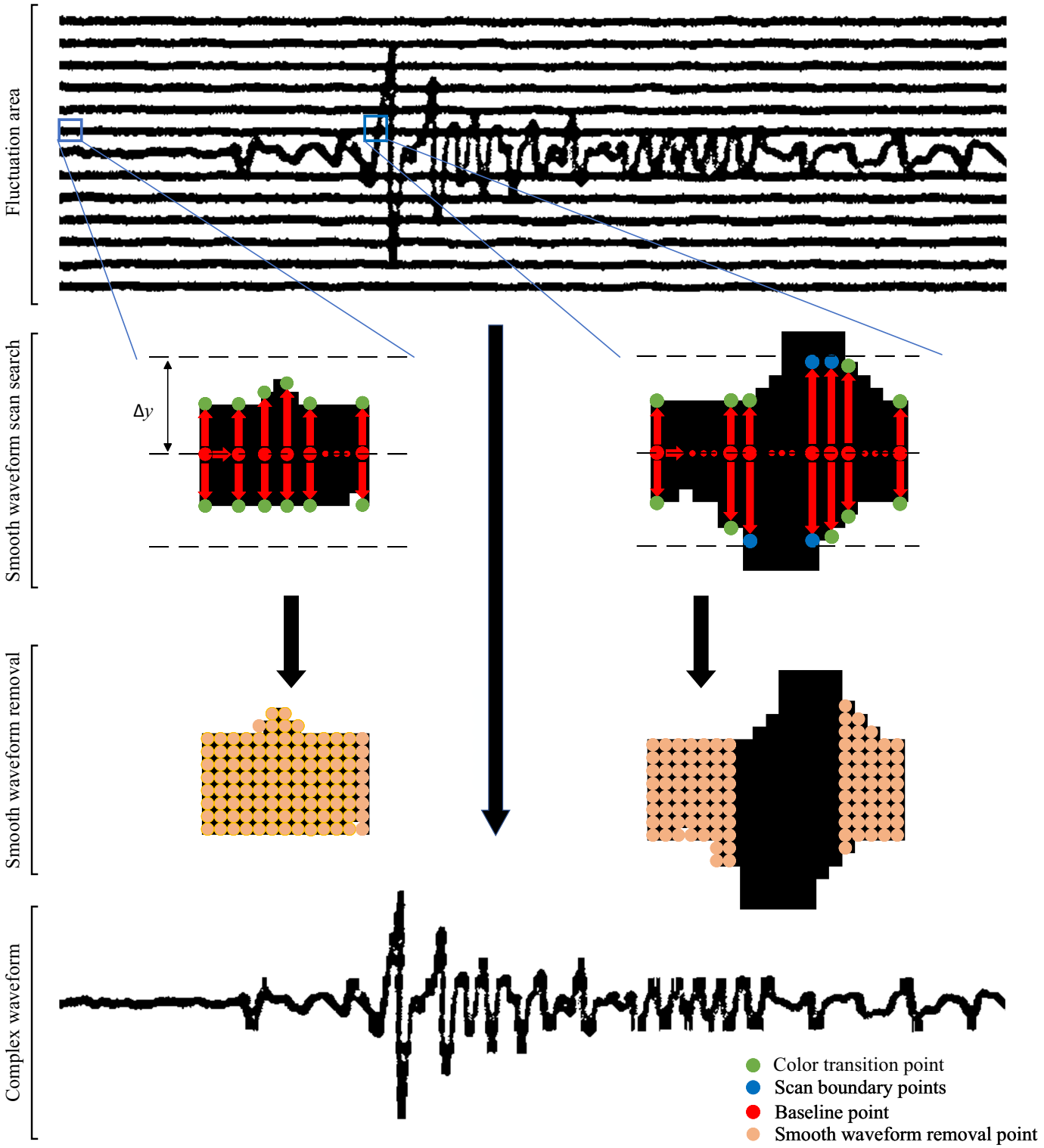

Figure 3. Smooth waveform removal algorithm description. The color version of this figure is available only in the electronic edition.

architecture improves hierarchical feature aggregation, enhancing the model's performance on resource-constrained devices. The architecture of YOLOv9 is divided into three parts: the backbone network, the neck network, and the head network. The backbone network is used for basic feature extraction; the neck network performs multiscale feature fusion; and the head network is responsible for object classification and localization.

We use YOLOv9s to detect the position of time markers, as shown in Figure 4a,c. We horizontally shift the coordinate point at the bottom left corner of the detection box to the right by Δy (Δy is the average width of the waveform) to obtain a new point,

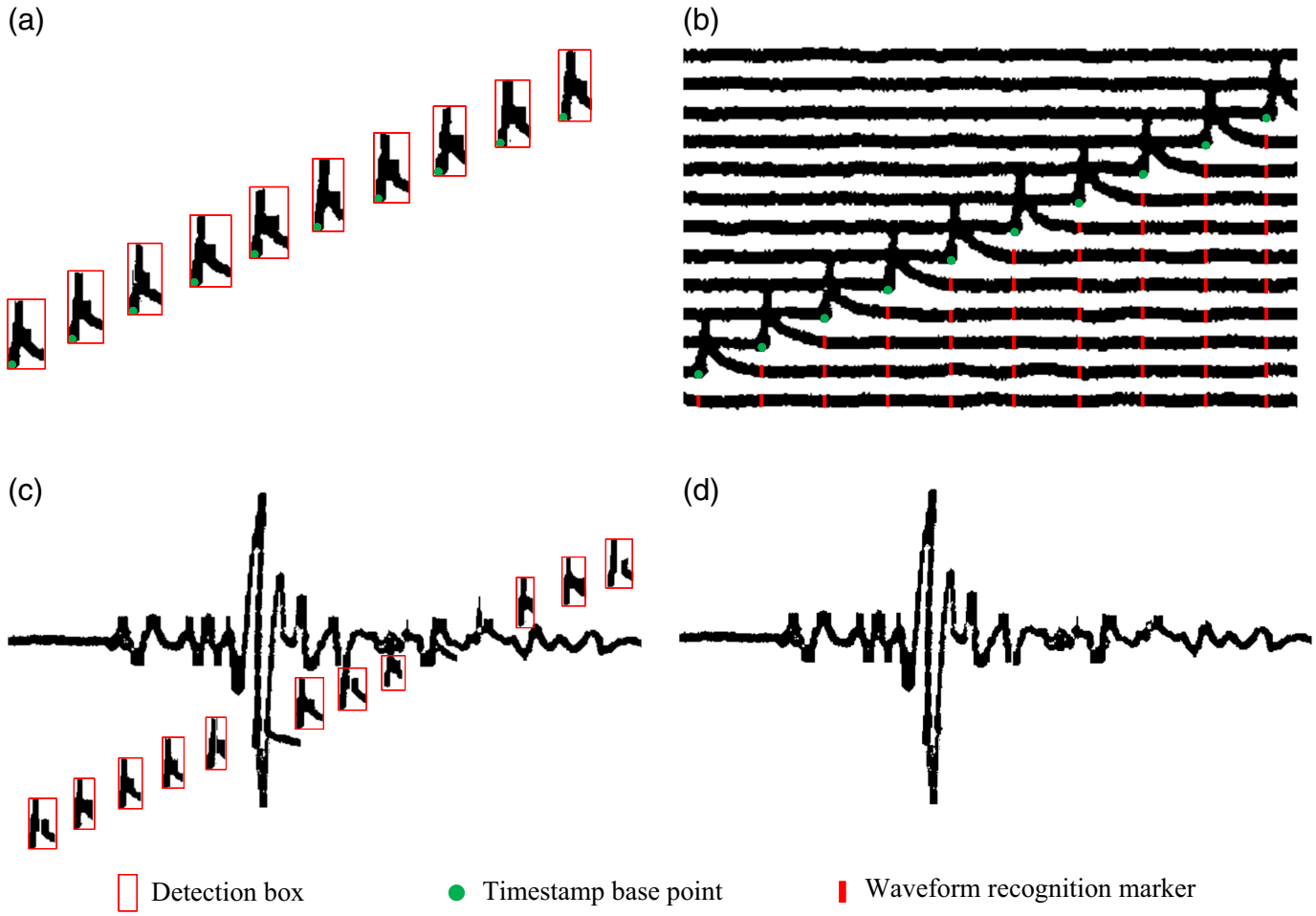

Figure 4. (a) Detect time markers and find timestamp base points. (b) Use the dynamic distributed waveform onset search algorithm to locate the waveform row number for each timestamp base point. (c) Detect time markers overlapping with complex waveforms. (d) After removing the time markers. The color version of this figure is available only in the electronic edition.

referred to as the timestamp base point (Fig. 4a). However, in rare cases for which the time markers are deeply overlapping with complex waveforms, detecting the positions of these time markers becomes difficult, and we need to manually mark the timestamp base points. Next, the dynamic distributed waveform onset search algorithm, previously mentioned, determines the row number of each timestamp base point. In subsequent steps, we will use the timestamp base point and its row number to calculate the arrival time of the waveform.

When time markers interfere with the extraction of complex waveforms, we remove them, as shown in Figure 4c,d. However, when time markers are deeply overlapping with complex waveforms, identifying and removing these markers becomes challenging. For these deeply overlapping time markers, manual removal is required. In addition, the previously mentioned smooth waveform removal algorithm will also partially remove time markers that are deeply overlapping with complex waveforms while removing the smooth waveforms.

We collected and manually labeled 270 seismograms for our dataset, which we divided into a training set (60%), a testing set (20%), and a validation set (20%). The yolov9s was trained on the training set, and its performance was evaluated on the testing set. In the testing set, the specific metrics are as follows: precision is 86.73%, recall is 81.67%, and the F1 score is 84.12%. These results indicate that the YOLOv9s has high accuracy and strong detection capability in the task of identifying time markers. Despite using a relatively small training set, the YOLOv9s model is still able to produce satisfactory results. In the future, as the size of the training dataset increases, the model's performance is expected to improve further. In addition, our research found that using deep learning models to detect types of time markers with distinct features is more effective. However, the model performs poorly for types of time markers with indistinct features or those similar to waveform features.

Waveform Digital Extraction and Reconstruction

Our strategy involves digitally extracting complex waveforms from analog seismograms that are structurally complex and of significant research importance while filling in smooth waveforms with corresponding line segments. This approach allows

us to focus on the features of the seismic data that are most likely to provide valuable insights into seismic activity, while effectively managing less informative segments by simplifying their representation.

Feature extraction of waveform peaks and troughs

The key to the digital extraction of complex waveforms lies in the feature extraction of peaks and troughs. Our method involves scanning each column of pixels in complex waveforms from top to bottom, starting from the top pixel $y = 0$ and scanning down to the bottom of the column at $y = H - 1$ (in which H is the height of the fluctuation area). We record the first color transition point in each column as the upper edge point of the waveform, and the last color transition point as the lower edge point, thus obtaining two sets of coordinates: one set describes the upper edge contour of the waveform, and the other set outlines the lower edge contour (Fig. 5a).

We use a combination of cubic spline interpolation and Gaussian filtering to process the waveform edge contours (Deriche, 1993; Boor, 2001). First, we use cubic spline interpolation to smooth and continuously process the entire edge contour based on the upper and lower edge coordinates, filling in gaps in the data to achieve a smoother and more coherent edge contour (Fig. 5b). This interpolation method is achieved by constructing a series of cubic polynomials (equation 16), interpolating between each pair of adjacent data points and satisfying a series of key conditions to ensure the continuity and smoothness of the curve. Specifically, the interpolation condition requires the curve to precisely match the data points (equation 17), the continuity condition ensures a smooth transition between data points (equation 18), and the natural boundary condition achieves a natural transition at the endpoints of the sequence (equation 19). Here, x_i and y_i are the coordinates of adjacent data points, and a_i, b_i, c_i, d_i are the coefficients of the curve in each interval.

$$S_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i, \tag{16}$$

$$S_i(x_i) = y_i, \quad S_i(x_{i+1}) = y_{i+1}, \tag{17}$$

$$S_i'(x_{i+1}) = S_{i+1}'(x_{i+1}), \quad S_i''(x_{i+1}) = S_{i+1}''(x_{i+1}), \tag{18}$$

$$S_0''(x_0) = 0, \quad S_{n-1}''(x_n) = 0. \tag{19}$$

However, due to the poor quality of the original seismograms, sampling errors, or factors introduced during the interpolation process itself, the interpolation results may contain slight fluctuations and noise. To effectively reduce these fluctuations while preserving important features and details of the edge contours, we use Gaussian filtering to further smooth the interpolated edge contours. The Gaussian filter uses the Gaussian function (equation 20) as a weighted average, applying a weighted average to each data point and its surrounding neighborhood, thereby producing smoother results.

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}, \tag{20}$$

in which x represents the offset of data points relative to the current processing point, and σ is the standard deviation of the Gaussian distribution that controls the range of the weight distribution. Choosing the appropriate σ value is crucial, a smaller σ preserves more details but provides less smoothing, whereas a larger σ produces smoother results but may blur important features. In this study, we chose $\sigma = 5$. Through this method, the Gaussian filtering further smooths the upper and lower edge contours (Fig. 5c), closely approximating the true boundaries of the original waveform.

Next, we search for peak and trough features based on the processed edge contours. Starting with the smoothed sequence of upper edge contours, we examine the y values of each nonboundary point (for each point $2 \le i \le n - 1$) to determine whether it is greater than its directly adjacent points. If a point's value is greater than that of its immediate left and right neighbors, it is considered a local maximum, that is, a peak point. Similarly, by traversing the lower edge contours for local minimums, trough points can be identified. Because data noise or minor fluctuations may produce insignificant peak points, we need to filter out peak points with significant features. To this end, we introduce a significance index P to quantify the importance of each peak point (Wasserstein and Lazar, 2016). For each peak point, its significance P is defined as $P = H_{\text{peak}} - H_{\text{baseline}}$, in which H_{peak} is the height of the peak point, and H_{baseline} is the height of the higher of the two local minimums found extending from the peak point to both sides along the upper edge contour. Only when the significance P of a peak point reaches at least the threshold T, do we consider the peak point significant. Trough points are also quantified for their importance using the significance index P, ensuring that the selected peak and trough points are worth searching. A smoothed upper-edge contour can be represented by the function $y(x)$, in which x represents points on the contour. For each significant peak point x_i, the determination of its adjacent inflection points on the left and right depends on the sign change of the second derivative of $y(x)$. Specifically, by examining the change in the sign of the second derivative to the left and right of x_i, we can determine the inflection points on both sides (Fig. 5d), as shown in equation (21). The points between the corresponding left and right inflection points x_{li} and x_{ri} for peak point x_i are combined to form the characteristic curve about each peak point. Similarly, the characteristic curves for each trough point can be obtained, and the characteristic curves of peak and trough points serve as features of the peaks and troughs (Fig. 5e).

$$\operatorname{sign}\left(\left|\left(\frac{\mathrm{d}^2y}{\mathrm{d}x^2}\right)|_{x-1}\right) \neq \operatorname{sign}\left(\left|\left(\frac{\mathrm{d}^2y}{\mathrm{d}x^2}\right)|_{x+1}\right). \tag{21}$$

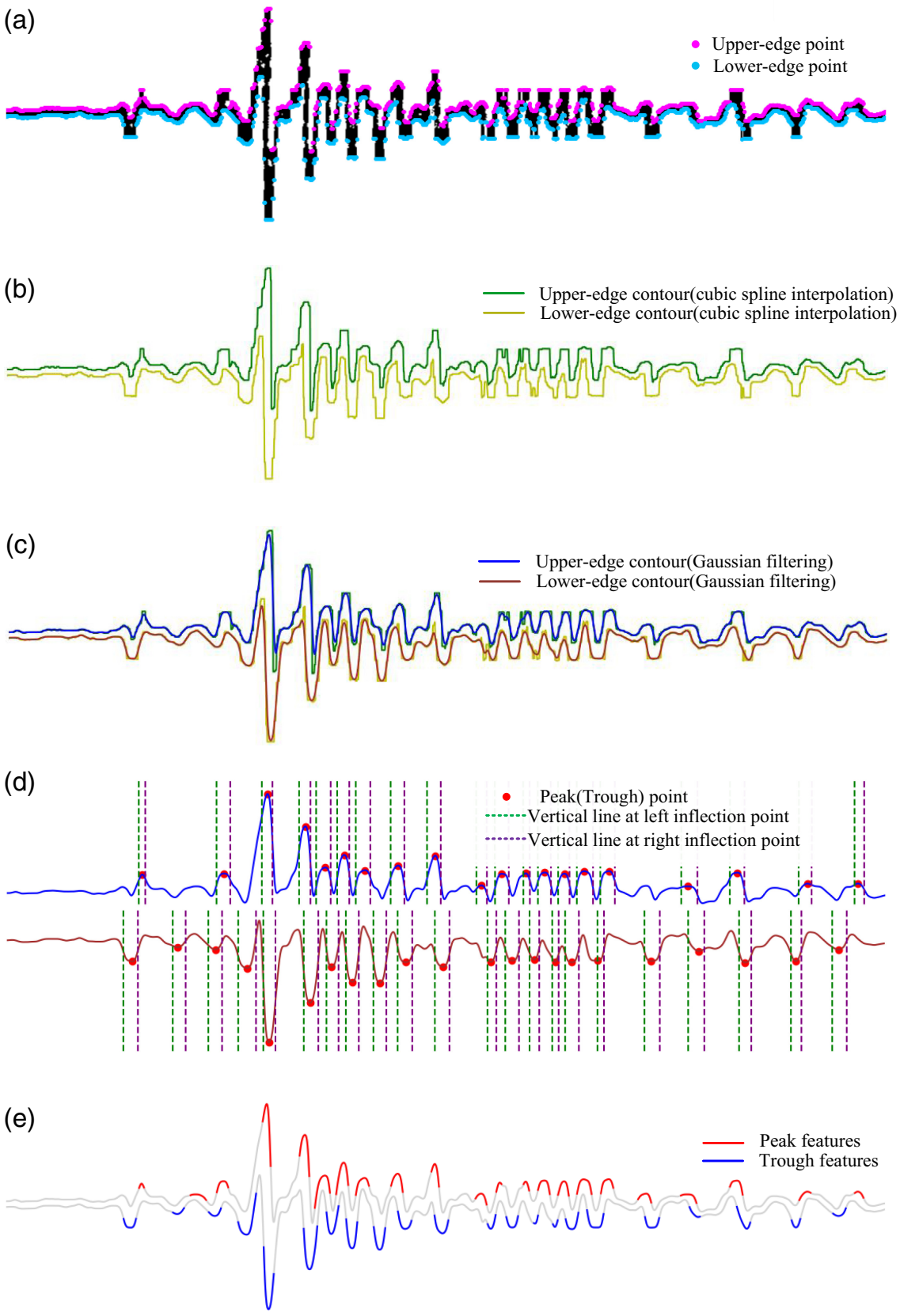

Figure 5. (a) Waveform edge contour extraction. (b) Perform cubic spline interpolation on the extracted waveform edge contours. (c) Apply Gaussian filtering to the edge contours interpolated by cubic spline. (d) Identify significant peaks and valleys and locate inflection points on their left and right sides. (e) Feature curves of peaks and troughs. The color version of this figure is available only in the electronic edition.

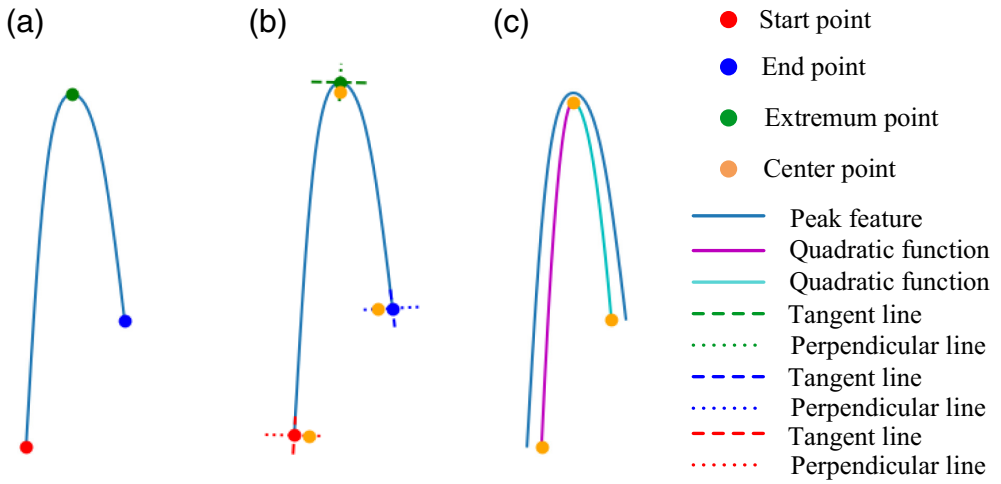

Figure 6. (a) Start point, endpoint, and extremum point of peak feature curve. (b) The tangent lines and perpendicular lines at the three points on the feature curve, and the center points corresponding to the three points. (c) Segmented quadratic function model that matches the peak and trough features in the actual waveform. The color version of this figure is available only in the electronic edition.

Waveform peak and trough restoration

Because we extract peak and valley features from the edge contours of waveforms, which are generated by pen strokes that have a certain width, the extracted peak and valley features are affected by the width of the pen strokes. This influence causes these features to inaccurately represent the actual waveform. Therefore, we apply geometric transformations to the peak and valley features to reduce errors caused by the pen stroke width during the waveform extraction process, thus better restoring the actual waveform.

We first locate the start point, end point, and extremum point of each peak feature curve as well as their center points relative to the pen width (Δy) (Fig. 6a). For the start and end points, we initially calculate the slope m of these points on the peak feature curve (equation 22) obtained by calculating the slope of the line segment formed by two adjacent points (x_1,y_1) and (x_2,y_2). Next, we determine the perpendicular normal slope $m_\perp$ (equation 23). Based on the normal slope $m_\perp$ and a given inward distance d ($d = \Delta y/2$), we calculate the coordinates of the point moved inward along the normal direction using the geometric principles of right triangles, with horizontal displacement Δx and vertical displacement Δy calculated using equations (24) and (25). These displacement values are then added to the coordinates of the starting and ending points to obtain the center coordinates of the start and end points of the peak feature curve. The same process is applied to obtain the center coordinates of the starting and ending points of the trough feature curve. For the extremum points in the peak features, they are directly moved upward by distance d to determine the center position of the peak. The extremum points in the trough features are moved downward by distance d to obtain the center point (Fig. 6b).

Next, we use these three center points to construct two segmented quadratic functions to accurately restore the peak (or trough) features in the actual waveform. First, the center of the extremum point serves as the vertex of the quadratic function, dividing the feature into two segments: one from the center of the strat point to the center of the extremum point, and the other from the center of the extremum point to the center of the end point. For each segment, we use the vertex form of the quadratic function $f(x) = a(x-h)^2 + k$, in which (h,k) are the coordinates of the center of the extremum point. We solve for the coefficient a for each segment to ensure that the quadratic function not only passes through the center of the extremum point but also precisely through the center of the start and end points. By establishing and solving a system of equations that includes these three center points, we obtain two coefficients a_1 and a_2, thus defining two continuous, smoothly transitioning segmented quadratic functions at the center point of the extremum. This way, we obtain a segmented quadratic function model that matches the actual waveform features of peaks and troughs in the actual waveform (Fig. 6c). Each method has its unique advantages. For processing periodic data, the sine function method is usually a good choice. However, for our specific need, which is to reduce the error caused by stroke width during waveform extraction, the quadratic function method has shown superior performance. Therefore, after evaluating various strategies, we have chosen to use the quadratic function method as our processing strategy. The peak and trough features mentioned in the following content refer to their quadratic function models.

$$m = \frac{y_2 - y_1}{x_2 - x_1}, \tag{22}$$

$$m_\perp = -\frac{1}{m}, \tag{23}$$

$$\Delta x = \sqrt{\frac{d^2}{1 + m_\perp^2}}, \tag{24}$$

$$\Delta y = m_\perp \times \Delta x. \tag{25}$$

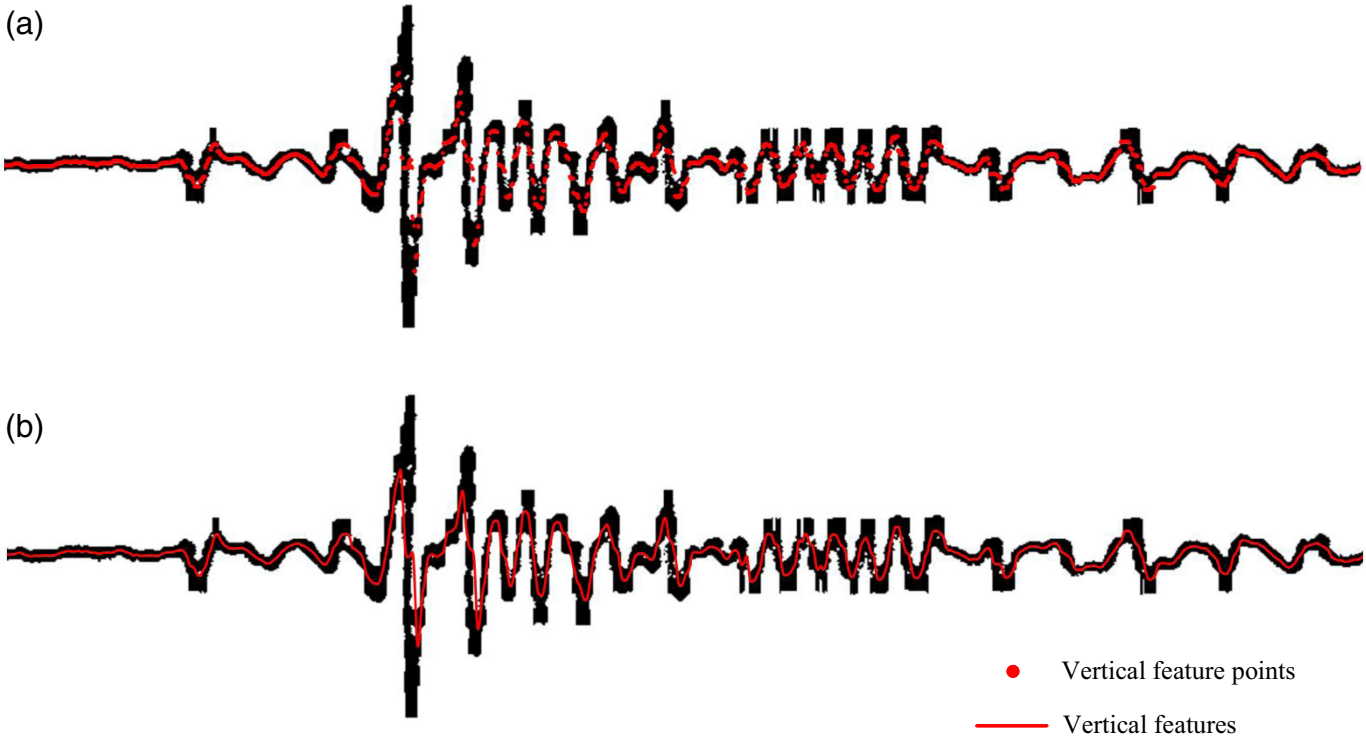

Figure 7. (a) Vertical features of the waveform. (b) Gaussian filtering to smooth. The color version of this figure is available only in the electronic edition.

Vertical feature extraction of waveforms

To extract vertical features from complex waveforms, we scan each column of pixels in the complex waveforms from top to bottom, recording the first color transition point in each column as the upper-edge point of the waveform y_{upper}, and the last color transition point as the lower edge point of the waveform y_{lower}. We calculate the arithmetic mean of y_{upper} and y_{lower} in each column as the feature extraction point y_{feature} for that column (equation 26). All the feature extraction points together represent the vertical features of the waveform (Fig. 7a).

Subsequently, these vertical feature points undergo cubic spline interpolation and Gaussian filtering to smooth and remove noise (Fig. 7b). The cubic spline interpolation provides a smooth, continuous curve that effectively models the vertical profile of the waveform, whereas the Gaussian filtering helps in reducing noise and minor fluctuations, ensuring that the extracted features are more representative of the actual waveform characteristics.

$$y_{\text{feature}} = \frac{y_{\text{upper}} + y_{\text{lower}}}{2}. \tag{26}$$

Complex waveform reconstruction

We use the peak and trough features extracted from complex waveforms in previous steps, along with the vertical features, to reconstruct the waveform (Fig. 8a). First, we identify the overlapping areas in the time series (x coordinate) between the vertical features and the peak and trough features, and remove the points within these overlapping areas from the vertical features. Then, for each pair of adjacent peak feature sequence p and trough feature sequence v, we define the boundaries of the overlap area as b_{start} and b_{end} (equations 27, 28), calculate the length of the overlap area L (equation 29), and assign $[b_{\text{start}}, b_{\text{start}} + \lfloor L/2 \rfloor - 1]$ to the earlier starting peak or trough sequence, and $[b_{\text{end}} - \lfloor L/2 \rfloor + 1, b_{\text{end}}]$ to the later starting peak or trough sequence. This method of reconstruction (Fig. 8b) ensures that the sequence of the waveform on the time axis is continuous, ensuring the accuracy of the waveform data.

$$b_{\text{start}} = \max(p_{\text{start}}, v_{\text{start}}), \tag{27}$$

$$b_{\text{end}} = \min(p_{\text{end}}, v_{\text{end}}), \tag{28}$$

$$L = \text{end} - \text{start}, \tag{29}$$

in which p_{start}, v_{start} are the x coordinates of the onset points of the peak and trough feature sequences, respectively, and p_{end}, v_{end} are the x coordinates of the endpoints of the peak and trough feature sequences, respectively. Figure 9 shows the results of the digital extraction and reconstruction of complex waveforms.

Waveform sequence integration and time mapping

We treat each waveform in the seismogram as a row and first perform a dynamic distributed waveform onset search across the

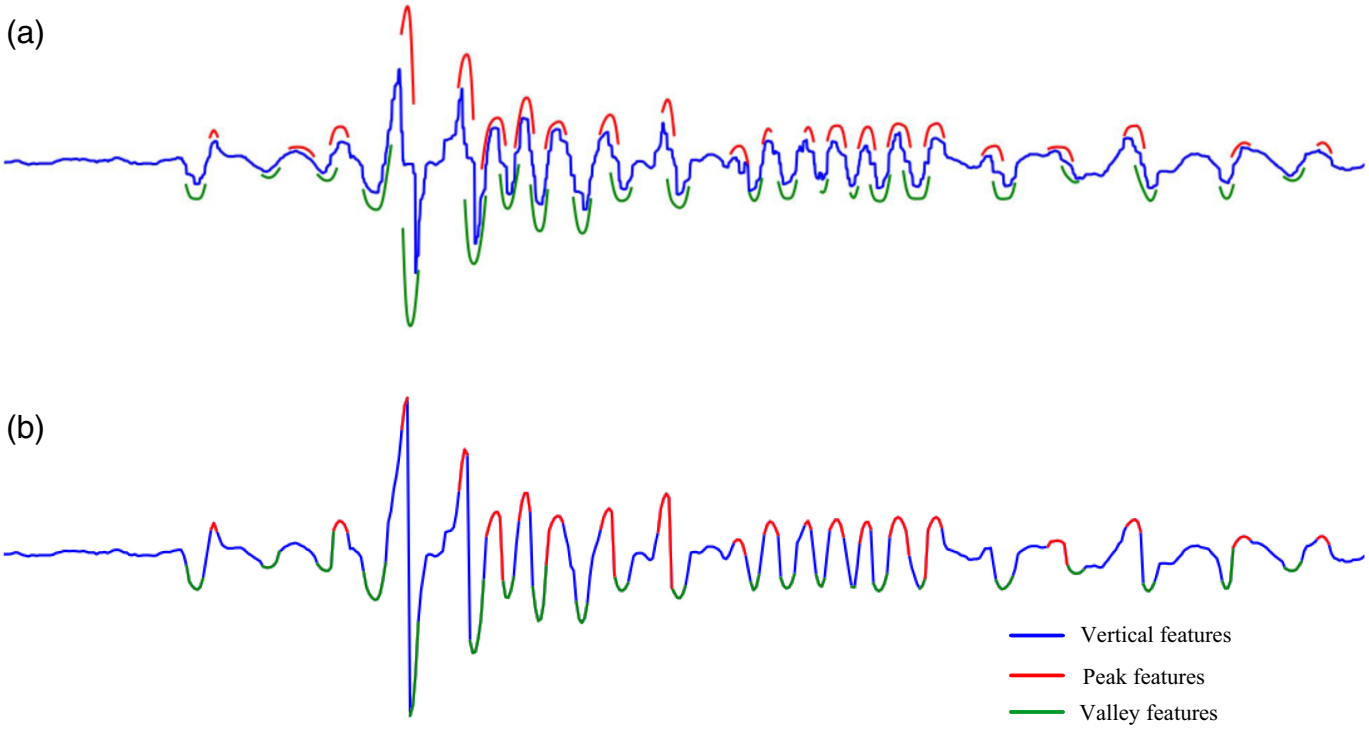

Figure 8. (a) Peak and trough features extracted from complex waveforms, along with the vertical features. (b) Reconstructed waveform. The color version of this figure is available only in the electronic edition.

entire seismogram to determine the total number of waveform rows N. During the process of framing and extracting complex waveforms, we carry out a dynamic distributed waveform onset search globally to lock the vertical position l (row number) of the complex waveforms. Next, we compute the sequence of smooth waveforms. The complete integration steps are as follows.

1. Once all complex waveforms in an seismogram have been extracted, sort all complex waveforms according to their row number l in the seismogram. Within the same row, the waveforms are further sorted by the x coordinate of their onset points.
2. Define the smooth waveform between the actual onset point of the seismogram and the onset point of the first complex waveform (x_1,y_1) as the initial smooth waveform I_{start} (equation 30). For any two adjacent complex waveforms A and B, define the smooth waveform between them as the buffer smooth waveform I_{AB} (equation 31), and the smooth waveform from the endpoint of the last complex waveform (x_n,y_n) to the actual endpoint of the seismogram as the ending smooth waveform I_{end} (equation 32).
3. Starting from the beginning, first insert I_{start}, and then sequentially concatenate each seismic waveform. Insert the calculated smooth waveform I_{AB} between every two adjacent waveforms and finally insert I_{end} (Fig. 10).

$$I_{\text{start}} = (l_1 - 1) \times L + x_1, \tag{30}$$

$$I_{AB} = (l_B - l_A) \times L - (x_{A\_\text{end}} - x_{B\text{start}}), \tag{31}$$

$$I_{\text{end}} = (N - l_n) \times L + (L - x_n), \tag{32}$$

in which $x_{A_{\text{end}}}$ is the x coordinate of the endpoint of waveform A, $x_{B_{\text{start}}}$ is the x coordinate of the onset point of waveform B, l_1 is the row number of the first complex waveform, and l_A and l_B are the row numbers of complex waveform A and B, respectively.

In the most well-preserved simulated seismograms, the explanatory table is usually preserved together with the waveform records. This explanatory table is mainly used to record the key parameters involved in the seismograms, such as the start time ($\text{Time}_{\text{start}}$) and the end time ($\text{Time}_{\text{end}}$) when the seismogram begins and ends recording. We use $\text{Time}_{\text{start}}$, Time_{end}, and the aforementioned timestamp base point to calculate the arrival time of each waveform point in the integrated waveform sequence.

First, we calculate the position x_i of each timestamp base point s_i in the integrated waveform sequence, as shown in the following equation:

$$x_i = (L_i - 1) \times L + X_i, \tag{33}$$

in which L_i and X_i represent the row number and horizontal coordinate of the timestamp base point s_i in the seismogram,

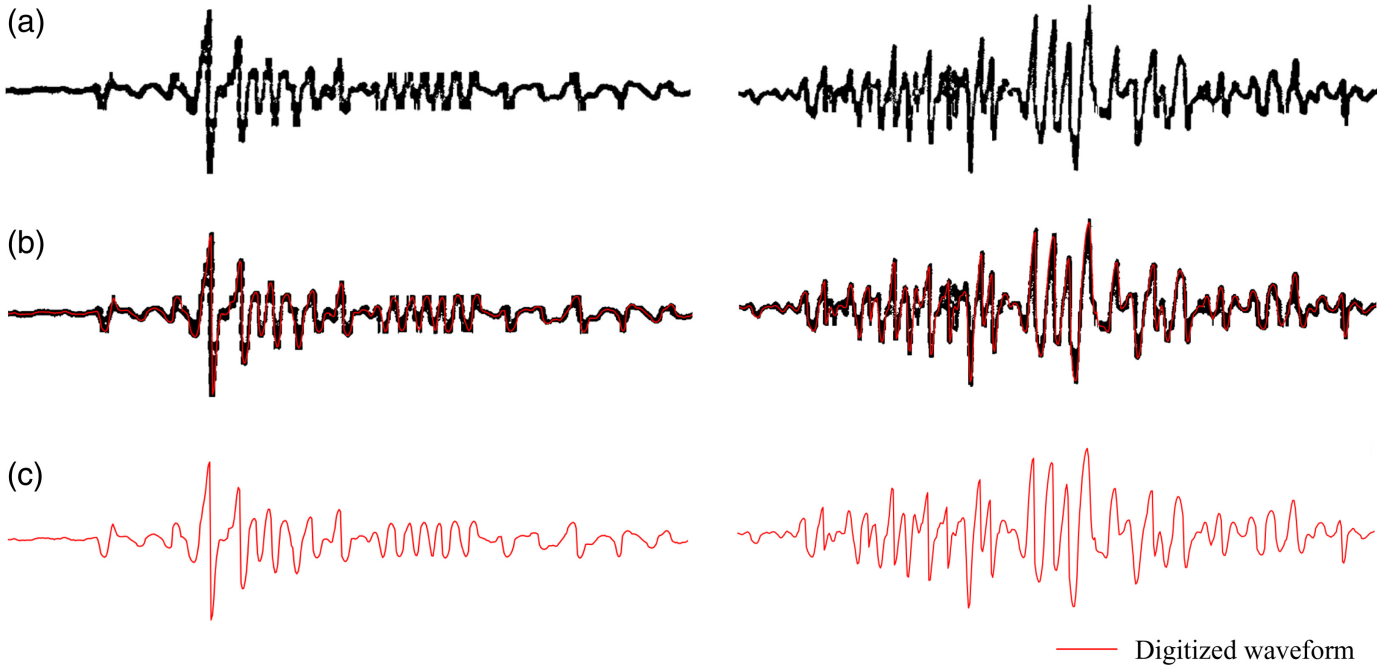

Figure 9. (a) Original waveform. (b) Overlay display of the original waveform and the digitally extracted waveform. (c) Digitally extracted waveform. The color version of this figure is available only in the electronic edition.

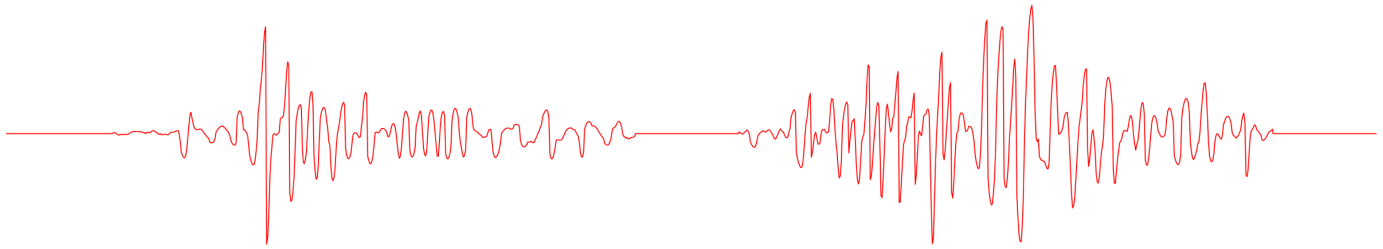

Figure 10. Integration effect diagram of the waveform time series. The color version of this figure is available only in the electronic edition.

respectively, and L is the length of each waveform in the seismogram.

Then, we calculate the arrival time of each waveform point. In the waveform sequence, the arrival time t of the waveform point x_i between timestamp base point s_i and s_{i+1} is calculated as shown in the following equation:

$$t = t_i + \left(\frac{x_i - s_i}{s_{i+1} - s_i}\right) \times \Delta t, \tag{34}$$

in which t_i is the arrival time of the timestamp base point s_i, and Δt is the time interval between timestamp base point. Similarly, we can also calculate the time of the waveform points between the start of the drawing and the first timestamp base point, as well as between the last timestamp base point and the end of the drawing.

Through the earlier methods, we can effectively integrate the waveforms in the seismograms and accurately map the arrival time to each waveform point. This not only facilitates the precise analysis of seismic waveforms but also provides a reliable data foundation for subsequent research and applications.

Conclusions

This study presents a novel method for the automated processing and digitization of analog seismograms, demonstrating its excellent performance in analog seismograms processing. First, we applied Otsu's method for threshold segmentation, calculating the grayscale histogram of the analog seismogram to find the threshold that maximizes interclass variance. This method divides the pixel values of the analog seismogram into foreground and background, improving processing accuracy. Next,

we implemented a dynamic distributed waveform onset-point-search method, which includes fuzzy search, outlier processing, and precise search phases. This method accurately locates the baseline onset point of each waveform and effectively identifies the total number of waveforms. For the diversity of waveforms in the seismograms, we designed a novel and efficient waveform classification algorithm that distinguishes between complex and smooth waveforms by scanning the number of white pixels along each waveform volatility midline. Furthermore, to calculate the arrival time of the waveform using time markers and to eliminate the interference of time markers in the extraction of complex waveforms, we utilized the YOLOv9s model to identify and remove these markers. Experiments show that our trained YOLOv9s model can produce acceptable results even with a smaller training set, and it is expected to improve with a larger training dataset. In addition, by scanning and processing the upper and lower edges of complex waveforms, we extracted the peak and trough features and performed geometric restoration. We also extracted the vertical features of the complex waveforms and accurately reconstructed the waveform structure based on the extracted features. Finally, we computed the sequence of smooth waveforms and integrated the time series of all waveforms, including both complex and smooth waveforms, ensuring the completeness and accuracy of the waveform data.

In summary, the method implemented in this study demonstrates significant advantages and stability in the digital processing of analog seismograms, potentially providing robust technical support for seismic research and data processing. However, the algorithm and system designed in this study have certain limitations when dealing with severe waveform distortions present in historical large earthquakes. Combining the recent advancements in meta-learning technology (Wang, Gong, *et al.*, 2024), we plan to further intelligently correct large waveform distortions to enhance the universality of our algorithm and system. In addition, we do not remove the pen curvature effect generated by the seismograph during the digitization of seismograms. Regarding the pen curvature effect, we plan to conduct in-depth research on correction algorithms to eliminate its impact on seismic waveforms, thereby improving the accuracy and reliability of the data. The accuracy of time marker recognition is also a focus of our future research.

In our study, we selected 500 seismograms from the Chengde seismic station in northern China, recorded in 1991, as our research subjects. These seismograms were captured using the DD-1 short-period seismic instrument—a Chinese-made device employing electronic amplification technology and ink pen for seismic waveform recording. In the same year, several earthquakes with magnitudes above 6.5 occurred in northern China. These paper-based seismograms were converted into raster images using a high-resolution scanner, with a scanning resolution of 600 DPI and a color depth of 24 bits, and saved in PNG format. In addition, the yolov9s model trained in this study, along with its source code, can be accessed at the GitHub repository (see Data and Resources), where subsequent related code and data will also be released.

Data and Resources

The YOLOv9s model is available at https://github.com/sandidi/v9s-TMS (last accessed July 2024). MATLAB can be accessed at www.mathworks.com/products/matlab (last accessed February 2024). There are no new data or resources to report for this article.

Declaration of Competing Interests

The authors acknowledge that there are no conflicts of interest recorded.

Acknowledgments

Maofa Wang and Fengshan Yang are contributed equally to this article. This work was supported by the National Natural Science Foundation of China (42164002) and Innovation Project of Guangxi Graduate Education (YCSW2023308). The authors would like to thank Editor-in-Chief Allison Bent, Managing Editor Miranda Bohl, and anonymous reviewers for their assistance in improving this article.

References

Bogiatzis, P., and M. Ishii (2016). Digitseis: A new digitization software for analog seismograms, *Seismol. Res. Lett.* **87,** no. 3, 726–736.

Boor, C. D. (2001). *A Practical Guide to Splines*, Springer, New York.

Bungum, H., C. D. Lindholm, and A. Dahle (2003). Long-period ground-motions for large European earthquakes, 1905-1992, and comparisons with stochastic predictions, *J. Seismol.* **7,** no. 3, 377–396.

Corona-Fernández, R. D., and M. Santoyo (2023). Re-examination of the 1928 Parral, Mexico earthquake (m6.3) using a new multiplatform graphical vectorization and correction software for legacy seismic data, *Geosci. Data J.* **10,** 178–192.

Deriche, R. (1993). Recursively implementing the Gaussian and its derivatives, *Technical Rept. RR-1893*, INRIA.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova (2019). Bert: Pre-training of deep bidirectional transformers for language understanding, J. Burstein, C. Doran, and T. Solorio (Editors), *Proc. of the 2019 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota, Association for Computational Linguistics, 4171–4186.

Draper, N. R. (2011). Review of the Cambridge dictionary of statistics, in *International Statistical Review*, Fourth Ed., B. S. Everitt and A. Skrondal (Editors), Vol. 79, Cambridge University Press, Cambridge, United Kingdom, 273–274.

Furumura, M., Y. Ogawa, K. Sakamoto, and R. S. Matsu'ura (2023). Automatic digitization of JMA strong-motion seismograms recorded on smoked paper: An attempt using deep learning. *Seismol. Res. Lett.* **94,** no. 6, 2712–2724.

He, K., X. Zhang, S. Ren, and J. Sun (2016). Deep residual learning for image recognition, *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 770–778.

Ishii, M., H. Ishii, B. Bernier, and E. Bulat (2014). Efforts to recover and digitize analog seismograms from Harvard-Adam Dziewonski observatory, *Seismol. Res. Lett.* **86,** no. 1, 255–261.

Leng, Z., M. Wang, Q. Wan, Y. Xu, B. Yan, and S. Sun (2024). Meta-learning of feature distribution alignment for enhanced feature sharing, *Knowl.-Based Syst.* **296,** 111875.

Okal, E. A. (2015). Historical seismograms: Preserving an endangered species, *GeoResJ* **6,** 53–64.

Otsu, N. (1979). A threshold selection method from gray-level histograms, *IEEE Trans. Syst. Man Cybern.* **9,** no. 1, 62–66.

Pintore, S., M. Quintiliani, and D. Franceschi (2005). Teseo: A vectoriser of historical seismograms, *Comput. Geosci.* **31,** no. 10, 1277–1285.

Rukstales, K. S., and M. D. Petersen (2019). Data release for 2018 update of the U.S. National Seismic Hazard Model, *U.S. Geol. Surv. data release*, doi: 10.5066/P9WT5OVB.

Stein, S., and M. Wysession (2003). *An Introduction to Seismology, Earthquakes, and Earth Structure*, Blackwell Publishing, Malden, Massachusetts.

Wang, C.-Y., I.-H. Yeh, and H.-Y. M. Liao (2024). Yolov9: Learning what you want to learn using programmable gradient information, available at http://arXiv.org/abs/2402.13616 (last accessed April 2024).

Wang, K., W. Ellsworth, G. C. Beroza, W. Zhu, and J. L. Rubinstein (2022). Develnet: Earthquake detection on develocorder films with deep learning: Application to the rangely earthquake control experiment, *Seismol. Res. Lett.* **93,** no. 5, 2515–2528.

Wang, K., W. L. Ellsworth, G. C. Beroza, G. Williams, M. Zhang, D. Schroeder, and J. Rubinstein (2018). Seismology with dark data: Image-based processing of analog records using machine learning for the rangely earthquake control experiment, *Seismol. Res. Lett.* **90,** no. 2A, 553–562.

Wang, M., Q. Gong, Q. Wan, Z. Leng, Y. Xu, B. Yan, H. Zhang, H. Huang, and S. Sun (2024). A fast interpretable adaptive meta-learning enhanced deep learning framework for diagnosis of diabetic retinopathy, *Expert Syst. Appl.* **244,** 123074.

Wang, M., Q. Jiang, Q. Liu, and M. Huang (2016). A new program on digitizing analog seismograms, *Comput. Geosci.* **93,** 70–76.

Wasserstein, R. L., and N. A. Lazar (2016). The ASA statement on p-values: Context, process, and purpose, *Am. Stat.* **70,** no. 2, 129–133.

Xu, Y., and T. Xu (2014). An interactive program on digitizing historical seismograms, *Comput. Geosci.* **63,** 88–95.

Manuscript received 25 May 2024
Published online 30 August 2024