scientific reports

OPEN

Pretreatment to terahertz absorption curves by narrow undulation constraint and Its quick implementation suggested by convex hull

Yizhang Li, Lingyu Liu, Ke Li, Zhongmin Wang✉, Tianying Chang & Wenqing Xu

In this work, a method of pretreating THz absorption curve is proposed, which leads to minimal range in absorption, reserving necessary undulation of curve for identification by convolutional neural network. The kernel thought of proposed method is about confining the undulation of curve with a pair of narrow parallel lines and solving their optimal position by consecutively rotation of normalized curve at two fixed points. A fast algorithm is further proposed based on features of convex hull, whose procedure is described in detail. The algorithm involves definition of some important point sets, calculating and comparing slopes and determining best choice out of 4 potential rotations. The rationality of searching critical point is illustrated in a geometric way. Additionally, the adaption of the method is discussed and real examples are given to show the capacity of method to extract nonlinear information of a curve. The study suggests that methods regarding computer graphics also contributes to feature extraction with respect to THz curve and pattern recognition.

Terahertz time domain spectroscopy is widely used for material detection and identification[1–4]. The curve of absorption or extinction coefficient is so related to the constituents of material that pattern recognition is conducted in various background[5–9]. Absorption peaks are not observed for pure substance with symmetrical molecular structure (for example, polyethylene). Besides, the peaks are less observable due to overlaps of component spectrum. Therefore, machine learning is significant for data mining in investigations involving but not limited to herbs, meats, tea, cereals according to previous reports. It is suggested that pretreatment to curves benefits model performance and reducing difficulty for training a satisfactory model. The conventional pretreatment includes Savitzky-Golay smoothing, filtering in frequency domain, multivariate scattering correction (MSC) and etc., which reserve essential feature for identification but adjust the value at every frequency sampling[10–12]. These algorithms assume the form of noise mathematically and all points are processed equally. However, the feature for identifying curves may be weakened and some parameter is empirically configured to obtain good results. In addition, methods involving computer graphics are seldom studied to bridge subsequent identification methods.

Convolutional neural network (CNN) has been employed to identify object in an image as an effective and a popular model[13–16]. When CNN is associated with a terahertz curve, a conversion (or mapping) from THz curve to image is necessary before model training. The spectrum curve is viewed as a meaningful boundary in image to separate the upper part and lower part, which however, are meaningless because no actual value hits them. As a result, every pixel in an image participates in the training of CNN model. Compressing the range in absorption for given frequency band would meet the expectation of reducing computing cost, whereas the difficulty is to reserve essential features for identification. As is shown in Fig. 1a, a schematic THz curve has a range that equals the difference of offset1 and offset2. Line 2 and Line 1 are parallel lines with frequency axis, which indicate the upper and the lower bound in absorption. If another two parallel lines are used to confine curve, the undulation (difference in Y offset) between them changes (as is shown in Fig. 1b and Fig. 1c). Thus, we seek optimal parallel lines to confine curve with minimal distance and carry out shear transform to generate an image that accommodates the proceeded curve with minimal redundancy in Y direction (absorption dimension). Enlighted by the basic thought, the proposed method is named narrow undulation constraint (NUC).

Institute of Automation, Key Laboratory of UWB & THz of Shandong Academy of Sciences, Qilu University of Technology (Shandong Academy of Sciences), Jinan, China. ✉email: 13969070215@163.com

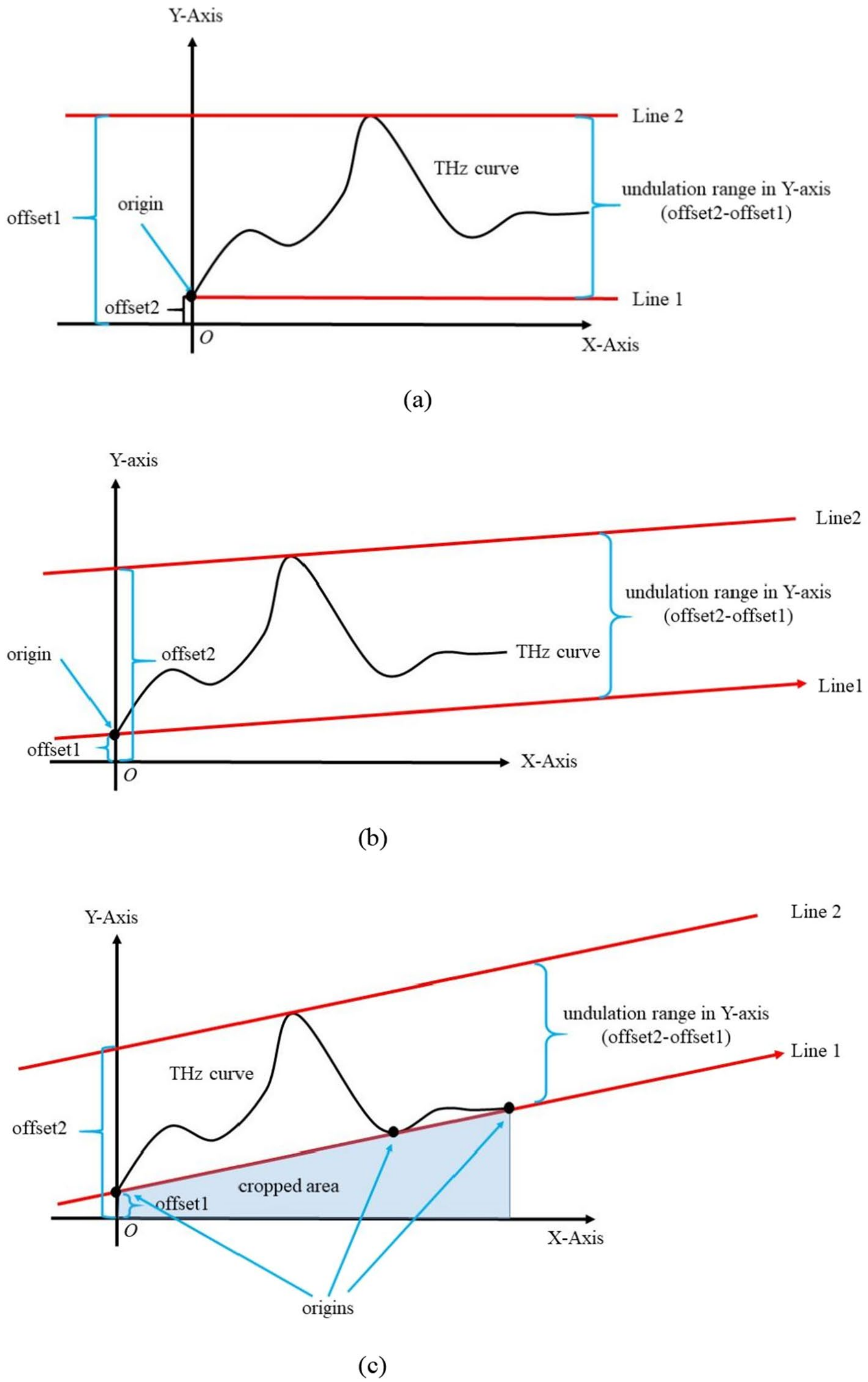

Figure 1. Schematic of Narrow Undulation Constraint: (**a**) original undulation in Y-axis; (**b**) adjusted undulation in Y-axis; (**c**) minimal undulation in Y-axis. Fig. 1 include 3 cases showing different undulation constraints. In Fig. 1a, the curve is confined by two lines parallel to X-axis (frequency axis). In this case, undulation in Y-axis direction equals the difference of maximum and minimum. In Fig. 1b, the curve is confined by two parallel lines and every line has one intersection point with curve. In this case, the undulation of curve in Y-axis direction can be further compressed if two parallel lines rotate around the present intersection points. In Fig. 1c, the curve is confined by two optimal parallel lines, of which one has one intersection point and the other has more than one intersection points with curve. In this case, the undulation of curve in Y-axis direction can't be compressed any more. Apply shear transformation to narrow band indicated by two parallel lines, the difference in Y offset of parallel lines turn to new range in Y axis, which allows fewer pixels to quantify change in absorption.

In this study we propose an algorithm of NUC based on convex hull to fulfill optimal shear whose graphic annotation is provided to strengthen understanding of mathematic expression. It is believed that mathematic definitions are important for describing procedure of algorithm precisely but the rationality behind formulations is better accomplished via illustrations. It is suggested that methods regarding computer graphics is complementary to algebraic method in adjusting shape of curves, which may have an impact on THz spectrum analysis and facilitate applications of CNN models in THz curve identification.

Method

Formulation. Assuming we have a curve with N effective samplings ($N \geq 3$) in frequency domain and the serial number of every point is determined by the ascending order of its frequency. The normalization is equivalent to a combination of translation and zooming. Then, the shear transformation is equivalent to cropping area beyond one of parallel lines by subtracting Y coordinates of one parallel lines from the normalized curve. Thus, we need matrices to denote input, translation, zoom factor, which are denoted by $\mathbf{X}_{\mathbf{N\times2}}$, $\mathbf{T}_{\mathbf{N\times2}}$, $\mathbf{Z}_{\mathbf{2\times2}}$ by (1), (2), (3) respectively where the subscript (or subscript in parenthesis) indicates the size of matrix. The final output of algorithm is denoted by Y in (4) and the range of its second column is expected to fall as much as possible, which is already described in introduction section. It is our goal to optimize $\mathbf{L}_{\mathrm{N\times2}}$, the matrix to denote cropping line.

$$\mathbf{X} = \begin{bmatrix} f_1 & value\left(f_1\right) \\ \vdots & \vdots \\ f_N & value\left(f_N\right) \end{bmatrix} \tag{1}$$

$$\mathbf{T} = \begin{bmatrix} -f_1 & -\min\left(value\right) \\ \vdots & \vdots \\ -f_1 & -\min\left(value\right) \end{bmatrix} \tag{2}$$

$$\mathbf{Z} = \begin{bmatrix} \frac{1}{f_N - f_1} & 0 \\ 0 & \frac{1}{max(value) - min(value)} \end{bmatrix} \tag{3}$$

$$\mathbf{Y} = ((\mathbf{X} + \mathbf{T})\mathbf{Z} - \mathbf{L})\mathbf{Z}^{-1} - \mathbf{T} \tag{4}$$

Fast algorithm. It's sensible to determine both the direction of cropping line and one point it goes through so as to have numeric expression of **L**. In other words, the optimization on L is equivalent to a task of point searching since the straight cropping line is determined by some critical points in the curve. Assume that the direction of cropping line is indicated by θ, the acute angle between X-axis and cropping line. If the rotation from X-axis to cropping line is done counter clockwise, $\theta > 0$; if the rotation is done clockwise, $\theta < 0$. It is reasonable to conclude that $-0.5\pi < \theta < 0.5\pi$ for all cases if cropping line are not orthogonal to X-axis.

The steps of algorithm are described in detail in this section and summarized in Fig. 2. Above all, the visualization of algorithm is discussed in next section.

Step 1: Organize potential critical points by sets.

The local minimum and local maximum of original curve at f_i are defined by (5) and (6) respectively. The bump point and the pit point at f_i are defined by (7) and (8) respectively. Build sets including A, A_1, A_2, B, B_1 and B_2 according to (9), (10), (11), (12), (13), (14) respectively. It is easy to prove that A_1, A_2, B_1 and B_2 are mutually exclusive; $\mathrm{A}_1 \subset \mathrm{A}$ and $\mathrm{B}_1 \subset \mathrm{B}$. For convenience, the element in set A and B are denoted by a and b respectively.

$$\left(f_i, value(f_i)\right) \text{ is a local minimum, s.t.} \begin{cases} value\left(f_i\right) < value\left(f_{i-1}\right) \\ value\left(f_i\right) < value\left(f_{i+1}\right) \end{cases}, 2 \leq \mathrm{i} \leq N-1 \tag{5}$$

$$\left(f_i, value(f_i)\right) \text{ is a local minimum, s.t.} \begin{cases} value\left(f_i\right) > value\left(f_{i-1}\right) \\ value\left(f_i\right) > value\left(f_{i+1}\right) \end{cases}, 2 \leq \mathrm{i} \leq N-1 \tag{6}$$

$$\left(f_i, value(f_i)\right) \text{ is a bump point}$$

$$\text{s.t.} \begin{cases} \left(value\left(f_i\right) - value\left(f_{i-1}\right)\right) \cdot \left(value\left(f_{i+1}\right) - value\left(f_i\right)\right) > 0 \\ value\left(f_{i+1}\right) - value\left(f_i\right) < value\left(f_i\right) - value\left(f_{i-1}\right) \end{cases} \tag{7}$$

$\left(f_i, value(f_i)\right)$ is a pit point,

$$\text{s.t.} \begin{cases} \left(value\left(f_i\right) - value\left(f_{i-1}\right)\right) \cdot \left(value\left(f_{i+1}\right) - value\left(f_i\right)\right) > 0 \\ value\left(f_{i+1}\right) - value\left(f_i\right) > value\left(f_i\right) - value\left(f_{i-1}\right) \end{cases} \tag{8}$$

$$\mathrm{A} = \{i | value\left(f_i\right) \geq value\left(f_j\right), 1 \leq i, j \leq N, i \text{ and } j \text{ are integers}\} \tag{9}$$

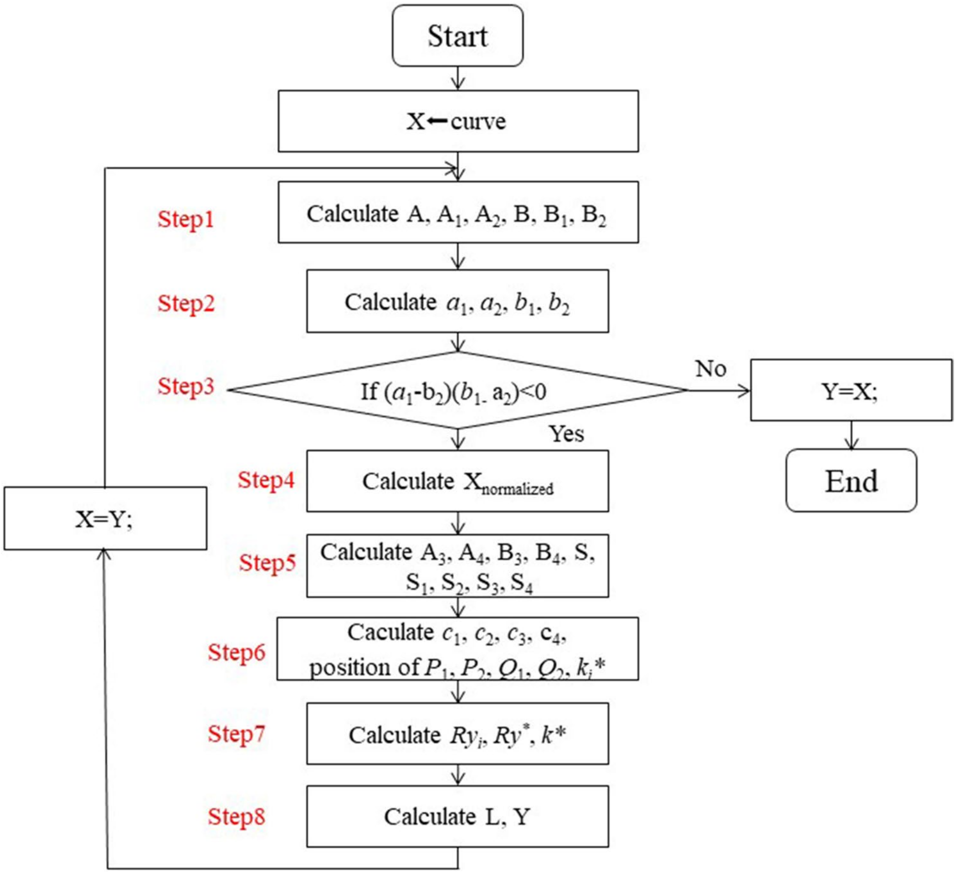
Start
X←curve
Step1
Calculate A, A_1, A_2, B, B_1, B_2
Step2
Calculate a_1, a_2, b_1, b_2
Step3
If $(a_1\text{-}b_2)(b_{1\text{-}}\ a_2)<0$
No
Y=X;
End
Yes
Step4
Calculate $X_{\text{normalized}}$
X=Y;
Step5
Calculate A_3, A_4, B_3, B_4, S, S_1, S_2, S_3, S_4
Step6
Caculate c_1, c_2, c_3, c_4, position of P_1, P_2, Q_1, Q_2, k_i*
Step7
Calculate Ry_i, Ry^*, k*
Step8
Calculate L, Y

Figure 2. Schematic of the fast algorithm. Fig. 2 consists of all steps described in fast algorithm section. Note that there is an 'if-else' statement in block diagram, whose effect is to check the adaption of method to current curve (discussed in detail in Discussion Part).

$$\mathrm{A}_1 = \{i|(f_i, value(f_i))\text{is a local maximum}\} \tag{10}$$

$$\mathrm{A}_2 = \{i|(f_i, value(f_i))\text{is a bump}\} \tag{11}$$

$$\mathrm{B} = \{i|value(f_i) \le \text{value}(f_j), 1 \le i, j \le N, i \text{ and } j \text{ are integers}\} \tag{12}$$

$$\mathrm{B}_1 = \{i|(f_i, value(f_i))\text{is a local minimum}\} \tag{13}$$

$$\mathrm{B}_2 = \{i|(f_i, value(f_i))\text{is a pit}\} \tag{14}$$

Step 2: Record indexes for potential rotation centers based on Step 1.

The calculation of a_1, a_2, b_1 and b_2 is shown in (15), (16), (17), (18) respectively. Particularly, the expression 'min' denotes the operation to get minimal value in the set while the expression 'max' denotes the operation to get maximal value in the set. P_1 and P_2 are both global maximum of the curve; Q_1 and Q_1 are both global minimum of the curve. In particular, P_1 and P_2 overlaps if $\text{Card}(A_1) = 1$, thus $a_1 = a_2$; Q_1 and Q_2 overlaps if $\text{Card}(B_1) = 1$, thus $b_1 = b_2$. The expression 'Card' means to obtain total number of elements in the set (cardinal number).

$$a_1 = \min(\mathrm{A}) \tag{15}$$

$$a_2 = \max(\mathrm{A}) \tag{16}$$

$$b_1 = \min(\mathrm{B}) \tag{17}$$

$$b_2 = \max(\mathrm{B}) \tag{18}$$

Step 3: Check if curve can be processed effectively by the algorithm.

Check if $(a_1-b_2)(b_1-a_2) < 0$. If this condition is met, the curve can be processed effectively. Just follow Step 4–8 and $\mathbf{Y} = \mathbf{X}$, then restart from Step 1. If $(a_1\text{-}b_2)(b_1\text{-}a_2) > 0$, the current $\mathbf{X}$ would be final output.

Step 4: Normalize original curve and obtain coordinate in Euclidean Space.

The position of every point in Euclidean Space after normalization can be determined by (19) where the definition of matrix Z is found in (3). The coordinate of normalized points is a function of serial number indicated by (20) and (21). After normalization, the curve is inlaid into a square area defined by {(0,0), (1,0), (1,1), (0,1)} in Euclidean Space whose edge equals one. The normalized curve has at least one intersection point with every edge. It is important that the type of one point does not change after normalization because the normalization have no impact on the sorting of values. For example, one point is known as a bump point in original coordinate and it's still a bump point after normalization although the coordinate values for two directions change. Such phenomena can be generalized to points marked as pit, local minimum, local maximum.

$$\mathbf{X}_{\mathbf{normalized}} = (\mathbf{X} + \mathbf{T})\mathbf{Z} \tag{19}$$

$$x(i) = \frac{f_i - f_1}{f_N - f_1} 1 \leq i \leq N, i \text{ is an integer} \tag{20}$$

$$y(i) = \frac{value(f_i) - value(f_b)}{value(f_a) - value(f_b)} 1 \leq i \leq N, i \text{ is an integer}, a \in \mathrm{A}, b \in \mathrm{B} \tag{21}$$

Obviously, according to our definition, we get $y(a_1) = y(a_2) = 1$ and $y(b_1) = y(b_2) = 0$.

Step 5: Calculate slope of probable edge of convex hull in Cartesian coordinate.

Four sets to accommodate potential edge point of convex hull are built according to (22), (23), (24) and (25), respectively. Set S, the union of 4 mutually exclusive subsets S_1, S_2, S_3 and S_4 (indicated by (26)), contains all the slope values for comparison. The specific definition of S_1, S_2, S_3 and S_4 is shown in (27), (28), (29) and (30) respectively. The slope between two points whose indexes are i and j is calculated according to (31) where $1 \leq i$, $j \leq N$, i and j are integers.

$$\mathrm{A}_3 = \{1\} \cup \{i|i \in (\mathrm{A}_1 \cup \mathrm{A}_2), i < a_1\} \tag{22}$$

$$\mathrm{A}_4 = \{N\} \cup \{i|i \in (\mathrm{A}_1 \cup \mathrm{A}_2), i > a_2\} \tag{23}$$

$$\mathrm{B}_3 = \{1\} \cup \{i|i \in (\mathrm{B}_1 \cup B_2), i < b_1\} \tag{24}$$

$$\mathrm{B}_4 = \{N\} \cup \{i|i \in (\mathrm{B}_1 \cup B_2), i > b_2\} \tag{25}$$

$$\mathrm{S} = \mathrm{S}_1 \cup \mathrm{S}_2 \cup \mathrm{S}_3 \cup \mathrm{S}_4 \tag{26}$$

$$\mathrm{S}_1 = \{\mathrm{s}|\mathrm{s} = |slope(i, a_1)|, i \in \mathrm{A}_3\} \tag{27}$$

$$\mathrm{S}_2 = \{\mathrm{s}|\mathrm{s} = |slope(i, a_2)|, i \in \mathrm{A}_4\} \tag{28}$$

$$\mathrm{S}_3 = \{\mathrm{s}|\mathrm{s} = |slope(i, b_1)|, i \in \mathrm{B}_3\} \tag{29}$$

$$\mathrm{S}_4 = \{\mathrm{s}|\mathrm{s} = |slope(i, b_2)|, i \in \mathrm{B}_4\} \tag{30}$$

$$slope(i,j) = \begin{cases} \frac{y(i)-y(j)}{x(i)-x(j)}, i \neq j \\ 0, i = j \end{cases} \tag{31}$$

Step 6: Mark potential rotation and critical points for distance check.

Serial number c_1, c_2, c_3 and c_4 are used to indicate points in the convex hull of the normalized curve, which play a role in further calculations. The stipulation of them are seen in (32), (33), (34) and (35) respectively.

$$c_1 = \min(\mathrm{C}_1), \text{s.t.} \mathrm{C}_1 = \left\{i|i \in \mathrm{A}_3, \left|slope(i, a_1)\right| = \min(\mathrm{S}_1)\right\} \tag{32}$$

$$c_2 = \max(\mathrm{C}_2), \text{s.t.} \mathrm{C}_2 = \left\{i|i \in \mathrm{A}_4, \left|slope(i, a_2)\right| = \min(\mathrm{S}_2)\right\} \tag{33}$$

$$c_3 = \min(\mathrm{C}_3), \text{s.t.} \mathrm{C}_3 = \left\{i|i \in \mathrm{B}_3, \left|slope(i, b_1)\right| = \min(\mathrm{S}_3)\right\} \tag{34}$$

$$c_4 = \max(\mathrm{C}_4), \text{s.t.} \mathrm{C}_4 = \left\{i|i \in \mathrm{B}_4, \left|slope(i, b_2)\right| = \min(\mathrm{S}_4)\right\} \tag{35}$$

The coordinate of P_1, P_2, Q_1, Q_2 are confirmed depending on the rotation angle θ and rotation center. Assume parallel lines are named line1 and line 2 according to Y offset (offset2 > offset1). A universal rule to claim P_1, P_2, Q_1, Q_2 is described as follow:

P_i denotes point (the fixed rotation center or the critical point to be found) in line 1 and Q_i denotes point (the fixed rotation center or the critical point to be found) in line 2. If critical edge of convex hull is determined by line 1 (in case 1 and case 2), Q_1 and Q_2 overlaps. If critical edge of convex hull is supposed to determined by line 2 (in case 3 and case 4), P_1 and P_2 overlaps. $(x(a_1), y(a_1))$ and $(x(b_2), y(b_2))$ are fixed as rotation center of line 2 and line 1, respectively in both case 1 and case 4. As a contrast, $(x(a_2), y(a_2))$ and $(x(b_1), y(b_1))$ are fixed as rotation center of line 2 and line 1, respectively in both case 2 and case 3. P_1 lies to the left of P_2 unless they overlap; Q_1 lies to the left of Q_2 unless they overlap.

A deduction of above-mentioned rule in 4 cases are described as follow:

Case 1 If *point* $(x(a_1), y(a_1))$ is fixed as rotation center, P_1 is defined as $P_1(x(c_1), y(c_1))$ and P_2 is defined as $P_2(x(a_1), y(a_1))$. Meanwhile, Q_2 is defined as $Q_2(x(b_2), y(b_2))$ and Q_1 overlaps with Q_2.

Case 2 If point $(x(a_2), y(a_2))$ is *fixed* as rotation center, P_1 is defined as $P_1(x(a_2), y(a_2))$ and P_2 is defined as $P_2(x(c_2), y(c_2))$. Meanwhile, Q_1 is defined as $Q_1(x(b_1), y(b_1))$ and Q_2 overlaps with Q_1.

Case 3 If point $(x(b_1), y(b_1))$ is fixed as rotation center, P_2 is defined as $P_2(x(a_2), y(a_2))$ and P_1 overlaps with P_2. *Meanwhile*, Q_1 is defined as $Q_1(x(c_3), y(c_3))$ and Q_2 is defined as $Q_2(x(b_1), y(b_1))$.

Case 4 If point $(x(b_2), y(b_2))$ is fixed as *rotation* center, P_1 is defined as $P_1(x(a_1), y(a_1))$ and P_2 overlaps with P_1. Meanwhile, Q_1 is defined as $Q_1(x(b_2), y(b_2))$ and Q_2 is defined as $Q_2(x(c_4), y(c_4))$.

Four possible rotation angles expressed by its tan function is found in (36).

$$k_i^* = \begin{cases} slope(c_1, a_1), i = 1 \\ slope(c_2, a_2), i = 2 \\ slope(c_3, b_1), i = 3 \\ slope(c_4, b_2), i = 4 \end{cases} \tag{36}$$

Step 7: Determine optimal rotation on account of slopes and distance in Y direction.

Assume that the convex hull of curve will be confined by a pair of parallel lines that penetrate at least three points in its edge. The line with minor offset is named Line 1 and the other is named Line 2. The difference between offset of Line 2 and Line 1 is formulated by (37). The optimal Ry^* is found in (38). Correspondingly, k^* is determined according to (39). After comparing Ry_i, only one case is reserved as the final result and the corresponding rotation centers are recorded.

$$Ry_i = \begin{cases} 1 + (b_2 - a_1) * min\{|k_1^*|, |k_4^*|\}, i = 1or4 \\ 1 + (a_2 - b_1) * min\left\{\left|k_2^*\right|, \left|k_3^*\right|\right\}, i = 2or3 \end{cases} \tag{37}$$

$$Ry^* = \min\left\{Ry_1, Ry_2, Ry_3, Ry_4\right\} \tag{38}$$

$$k^* = k_i^* \text{s.t.} \begin{cases} Ry^* = Ry_i, Ry^* = Ry_j \\ \left|k_i^*| \le |k_j^*\right| \end{cases}, i, j \in \{1, 2, 3, 4\}, i < j \tag{39}$$

Step 8: Adjust curve in original space according to previous calculation.

Line 1 is denoted by **L** in matrix. The offset of Line 1 in Y axis is calculated by (40). The equation of Line for the investigated internal is formulated according to (41). After obtaining **L**, we get the adjusted curve according to (4).

$$offset_1 = \begin{cases} -b_2 k^*, k^* = k_1^* \\ -b_1 k^*, k^* = k_2^* \\ -b_1 k^*, k^* = k_3^* \\ -b_2 k^*, k^* = k_4^* \end{cases} \tag{40}$$

$$\mathbf{L} = \begin{bmatrix} 0 & k^* x_1 + offset_1 \\ \vdots & \vdots \\ 0 & k^* x_N + offset_1 \end{bmatrix} \tag{41}$$

Geometric Interpretation. Given that the meaning of matrices, points and sets is abstract, we show how the algorithm is designed with help of figures. It's necessary to claim that the 'curve' shall be understood as 'polylines' because of sampling and that a true curve does not exist in digital system. The polylines look as smooth as a curve if the change is not drastic in scale of sampling.

Assume there is a set of points S, and convex hull is the intersection of all convex sets containing S[17–19], which has been widely used in computer graphics. In a plane, a convex hull is the polygon with minimal area to cover all points. Some widely used convex hull algorithms are gift wrapping[20], Graham scan[21], quick hull[22], divide and

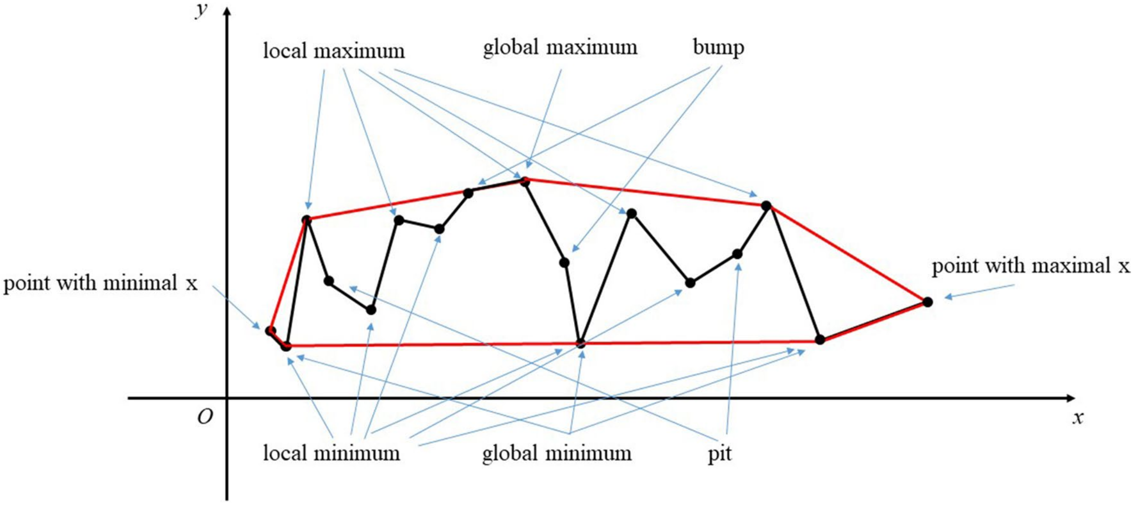

Figure 3. Schematic of convex hull and critical points defined in this paper. Fig. 3 gives examples of bump, pit, local maximum, local minimum, global maximum, global minimum. Point with maximal x and minimal x (x can be replaced by serial number) set the right and left boundary of curve, respectively. In addition, points with consecutive serial number but same y value do not belong to any above-mentioned point set, which are not depicted in this figure. They are not employed in this algorithm and do not affect result either.

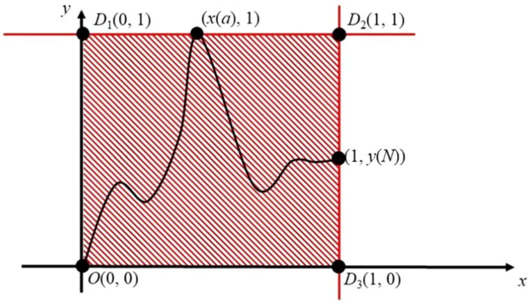

Figure 4. Schematic of normalized curve in a square. In Fig. 4, the normalized curve is laid in a square which is defined by {(0,0), (0,1), (1,1), (1,0)}. The curve has one intersection point with OD_1 and D_2D_3 respectively regardless of curve shape although the intersection point can be any of O(0,0), D1(0,1), D_2(1,1) and D_3(1,0). However, with the change of curve, one or more intersection points are allowed to find in D_1D_2 and OD_3.

conquer[23] and monotone chain[24]. A schematic of convex hull is presented in Fig. 3 that has 7 segments as edges in red. The example curve is connected by 16 segments in black and three of them overlaps with the edge of convex hull. According to local variation of curve, we build point set A, A_1, A_2, B, B_1, B_2 according to expressions in (9)–(14). The representative points are marked using blue arrows. The direction change between two adjacent edges of convex hull is not allowed to increase further for given set S. As a result, gift wrapping algorithm is designed to obtain convex hull. In Andrew's Algorithm[24], the search of convex hull starts from point with extreme X or Y value, we develop this thought to process normalized curve, where both dimension difference and range difference in X and Y direction are removed. In Fig. 4, the normalized curve has intersections with segment D_1D_2 and OD_3 that indicate the two parallel lines to confine undulation presently. In addition, the convex hull of polyline is a polygon that now locates in the square $OD_1D_2D_3$. In our work, we make use of convex hull to find optimal adjustment of curves by searching critical edges, however, we do not need to obtain entire convex hull. As a result, the proposed method is aimed at calculate convex hull.

As is mentioned, two parallel lines for reference are employed to confine undulation and one of them would coincide with one edge of convex hull since the continuous rotation of reference line centered at some point is like wrapping in Andrew's Algorithm. In our algorithm, the parallel line determined by convex hull is described as 'dominating parallel line' and the other as 'following parallel line'. The difference in offset of two parallel lines is the undulation we try to confine. Above all, to ensure wrapping is valid, we start from points with extreme Y values including $(x(a_1),1)$, $(x(a_2),1)$, $(x(b_2),0)$ and $(x(b_1),0)$ as fixed rotation center. The details are illustrated in Fig. 5 in 4 cases that correspond with the descriptions in 'Fast Algorithm' part.

In Fig. 5, the area of convex hull is depicted using a polygon in gray and the dominating parallel line is highlighted in blue. As the rotation direction is limited in 4 cases, we search another point to determine dominating parallel line by calculating the slope of line that penetrates $(x(a_1),1)$, $(x(a_2),1)$, $(x(b_2),0)$ and $(x(b_1),0)$, respectively. Due to the particularity of location of the fixed rotation centers, only a part of points is utilized, which reduces the calculation cost. The rejection subcases with respect to case 1 and case 3 are illustrated in Fig. 6, and the slope

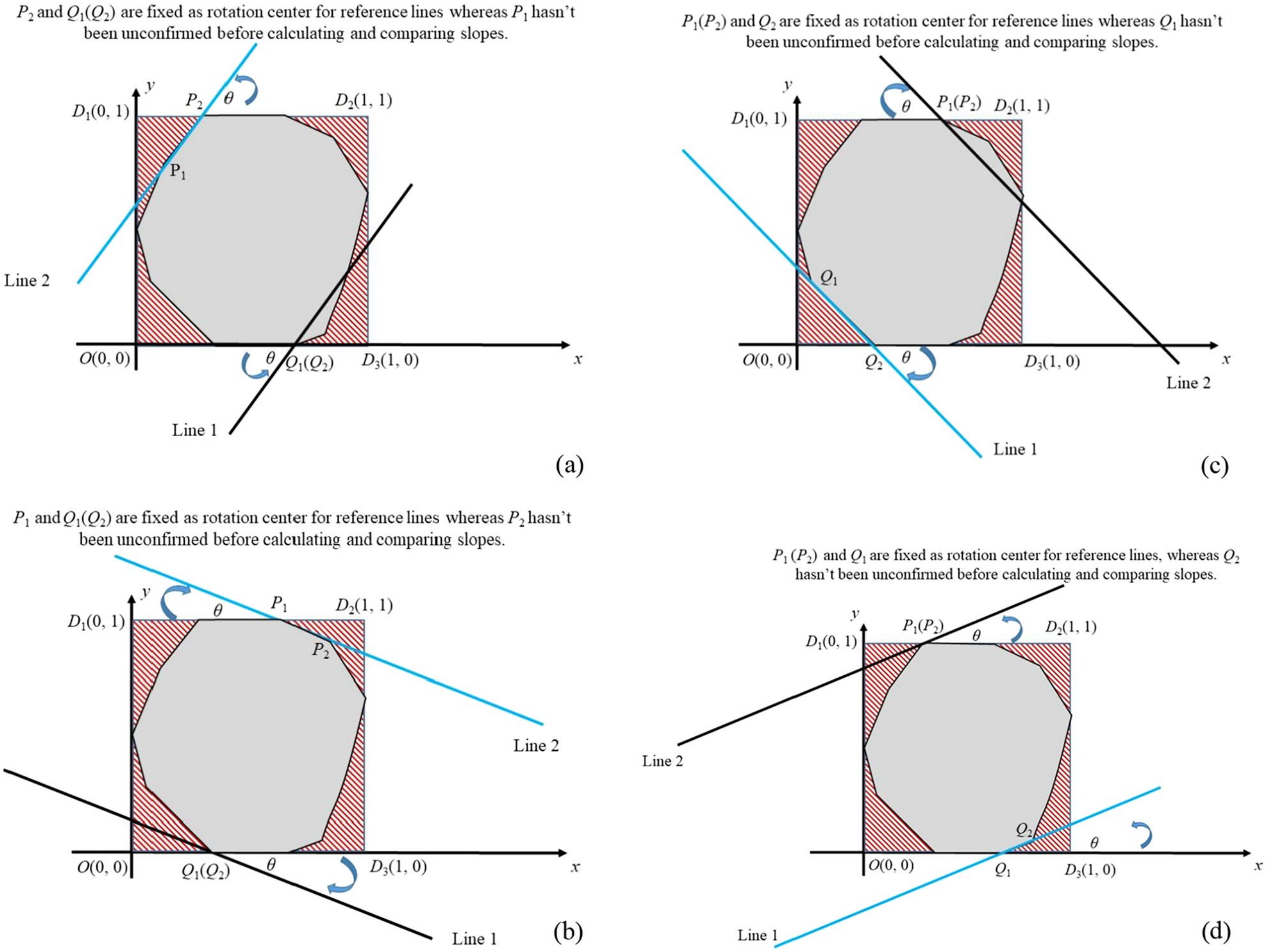
P_2 and $Q_1(Q_2)$ are fixed as rotation center for reference lines whereas P_1 hasn't been unconfirmed before calculating and comparing slopes.
y
$D_1(0, 1)$
P_2
θ
$D_2(1, 1)$
P_1
Line 2
$O(0, 0)$
θ
$Q_1(Q_2)$
$D_3(1, 0)$
x
Line 1
(a)
$P_1(P_2)$ and Q_2 are fixed as rotation center for reference lines whereas Q_1 hasn't been unconfirmed before calculating and comparing slopes.
y
$D_1(0, 1)$
θ
$P_1(P_2)$
$D_2(1, 1)$
Q_1
$O(0, 0)$
Q_2
θ
$D_3(1, 0)$
x
Line 2
Line 1
(c)
P_1 and $Q_1(Q_2)$ are fixed as rotation center for reference lines whereas P_2 hasn't been unconfirmed before calculating and comparing slopes.
y
$D_1(0, 1)$
θ
P_1
$D_2(1, 1)$
P_2
Line 2
$O(0, 0)$
$Q_1(Q_2)$
θ
$D_3(1, 0)$
x
Line 1
(b)
P_1 (P_2) and Q_1 are fixed as rotation center for reference lines, whereas Q_2 hasn't been unconfirmed before calculating and comparing slopes.
y
$P_1(P_2)$
$D_1(0, 1)$
θ
$D_2(1, 1)$
Line 2
Q_2
θ
$O(0, 0)$
Q_1
$D_3(1, 0)$
x
Line 1
(d)

Figure 5. Schematic of searching critical edge of convex hull. Fig. 5 involves 4 cases to search critical edge of convex hull, which starts by rotating line that go through the point in northwest, northeast, southwest, southeast as the rotation center, respectively. The gray polygon indicates the area that a typical convex hull covers. Imagine θ changes continuously from 0 for both parallel lines and if $|\theta|$ is big enough, a region formed by one of parallel lines and part of convex hull is found, proving that at least one vertice of convex hull lie beyond the narrow band. As the processed undulation change monotonically with θ, we just calculate extreme cases and select the optimal one out of four. One critical point to determine parallel lines belong to set A_3 A_4 B_3 B_4 respectively in 4 cases, respectively. The other critical point is $(x(a_1),1)$, $(x(a_2),1)$ $(x(b_2),0)$ and $(x(b_1),0)$ for 4 cases.

calculation is omitted for the points mentioned. Therefore, we conduct set operations in Eq. (22) and Eq. (24). Similarly, some points can also be filtered before calculating slope in case 2 and case 4, which corresponds with regulations in Eq. (23) and Eq. (25). As an exception, the slope of lines parallel with Y axis are defined as 0, which does not conform to corresponding mathematic term. Once the cardinal number of A_3, A_4, B_3, B_4 reaches 1, the index '*slope*' would be a number defined by one point itself. The treatment in Eq. (31) is to ensure robustness of algorithm.(Fig. 7).

However, confine all points within band determined by convex hull is not always successful if only one parallel line is considered (in Fig. 5a and Fig. 5c, at least one vertice of convex hull lies beyond the band). Thus, we need to select the appropriate rotation (indicated by k^*) from candidate ones ($\underline{k}_i^*$). The undulation in Y-axis between parallel lines (offset difference in Y-axis, denoted by Ry_i) is derived according to Fig. 8. Note that the difference between Ry_i and 1 is proportional to the absolute tan of angle. Thus, the four Ry_i indicates the utmost allowed rotation. Calculate Ry^* and k^* according to (38), (39) before deriving L based on slope and one point it goes through. In case 1 and case 4, the rotation center of line with minor offset (cropping line, line 1) in Y-axis is marked Q_1; in case 2 and case 3, the rotation center of cropping line in Y-axis is marked Q_2. After obtaining **L**, the curve is normalized reversely according to (4) and the range of undulation is reduced.

Results

An example based on a random curve (polylines made up by 21 points) is shown in Fig. 9 using (42) where the term 'rand' means a random value within interval [0,1]. As is seen, the fluctuation attached to rising trend is extracted and the resultant range is confined within local undulation indicated by random function. Note that the proposed method is not aimed at removing straight base line because the value of proceeded curve at every frequency sampling is no longer the random values employed in original **X**.

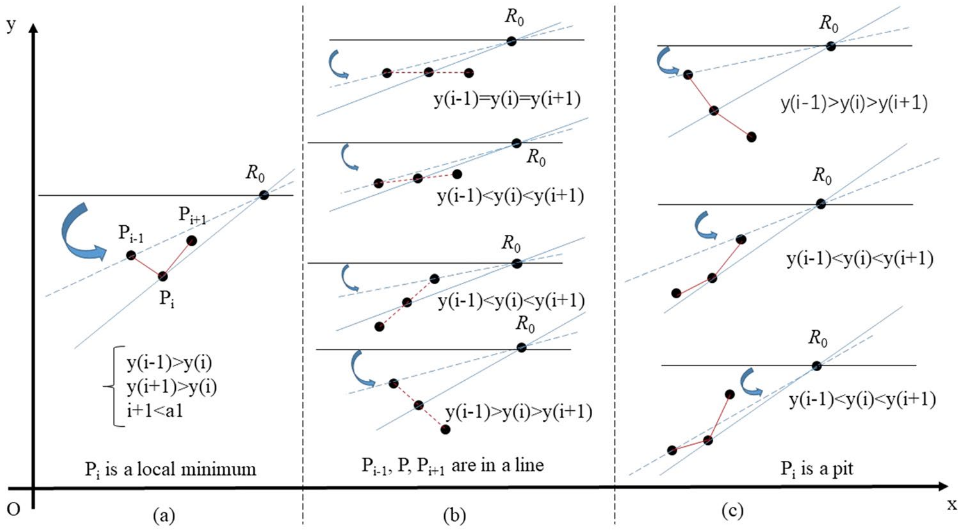

Figure 6. Rejection subcases in searching critical points: rotating line with $0 < \theta < 0.5\pi$ in case 1 and case 4: (**a**) any local minimum (**b**) any internal point aligned in curve (**c**) any pit. Fig. 6 shows why some points are ignored during calculating slopes. According to Fig. 6, with the increasing of θ, solid black line would coincide with blue dash line first and with blue solid line then, indicating P_i loses its qualification as a vertice in convex hull of curve. As a special case where red dash line coincides with blue dash line (not depicted), P_{i-1}, P_i, P_{i+1} would contribute to the same k_1^* and P_i is also not a vertice in convex hull.

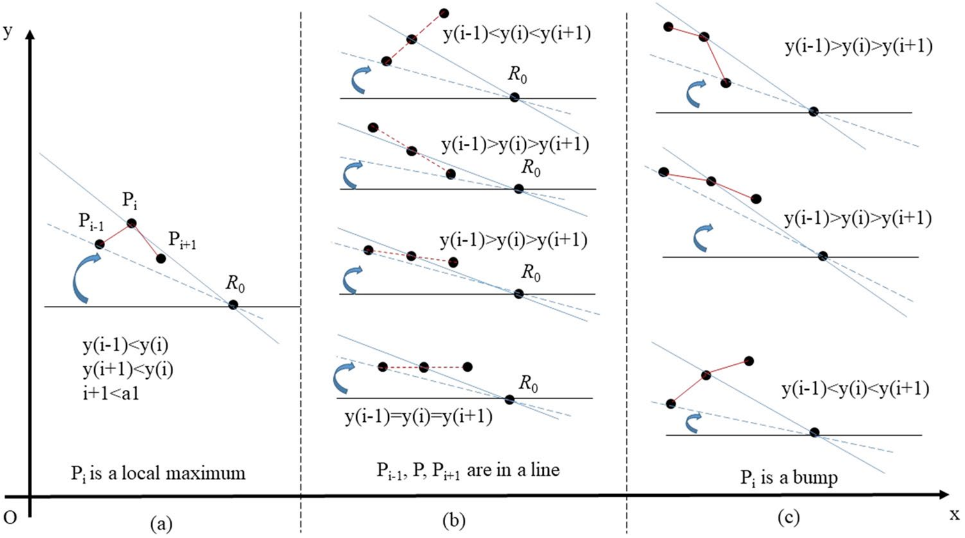

Figure 7. Rejection subcases in searching critical points: rotating line with $-0.5\pi < \theta < 0$ in case 2 and case 3: (**a**) any local maximum (**b**) any internal point aligned in curve (**c**) any bump. According to Fig. 7, with the decreasing of θ, solid black line would coincide with blue dash line first and with blue solid line then, indicating P_i loses its qualification as a vertice in convex hull of curve. As a special case where red dash line coincides with blue dash line (not depicted), P_{i-1}, P_i, P_{i+1} would contribute to the same k_3^* and P_i is also not a vertice in convex hull.

Examples based on 3 real THz curves (extinction coefficient of colla corii asini, a well-known Chinese traditional medicine) are shown in Fig. 10 to show cases with more points. Given complex chemical constituents of individual samples and test errors, 3 samples have similar profiles but different details. As is seen, the range in Y axis of all curves are notably reduced after processing, and the difference between curves are magnified. Thus, more pixels can be employed to reflect detailed difference rather than supplementary area. Besides, the Y value of processed curve would be no less than 0 and the minimum of processed curve in Fig. 10b is marked by an arrow.

Although THz spectrum is predicted as a fingerprint spectrum, notable peaks are not observed for many cases, especially in tests on mixture as the spectra of various components overlap. As the example mentioned, colla corii asini is also named Ejiao in China, which is produced by boiling donkey hide as well as auxiliary materials and the

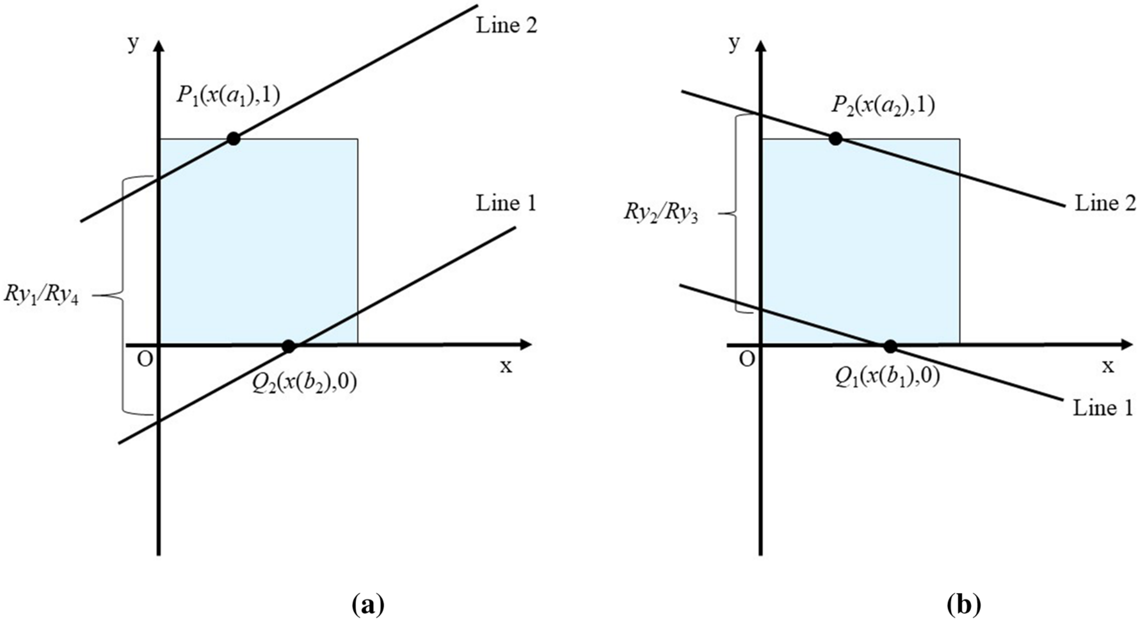

Figure 8. Calculation of distance between parallel lines in Y-axis: (**a**) Ry_1 or Ry_4 (**b**) Ry_2 or Ry_3. The calculation of Ry_i in Case 1 and case 4 (i.e., $i = 1$ or 4) is shown in Fig. 8a ; the calculation of Ry_i in Case 2 and case 3 (i.e., $i = 2$ or 3) is shown in Fig. 8b. It is easy to get (37) with elementary geometry and trigonometric function.

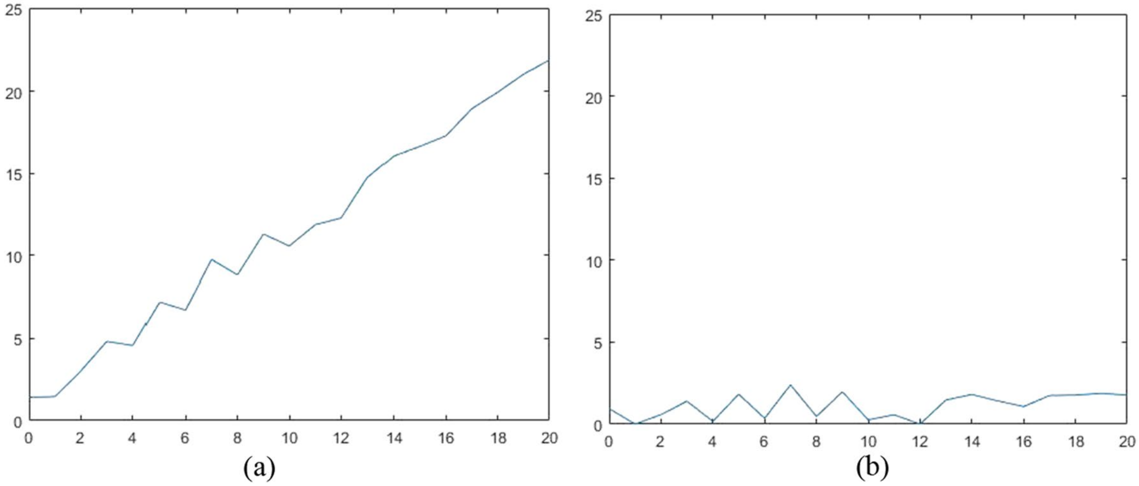

Figure 9. An NUC example based on simulated curve: (**a**) plot or simulated curve (**b**) plot of processed curve. The simulated curve is about a straight line with addictive white noise. After executing NUC, a similar noise figure is extracted for further study. Note that the algorithm is not used to remove line function but to highlight undulation.

chemical compound is very complex. A number of researchers focus on identify mixture without notable feature in THz band. NUC makes it possible to extract undulation for identification while removing overall linear trend that contributes to large range in absorption, therefore reducing pixels required for CNN model. An estimation can be made that original curve with 50 frequency samplings and absorption range of 0.045 need $50 \times 450 = 22500$ pixels according to Fig. 10a if range of 0.01 in absorption is quantified using 100 pixels. However, with the same ration of value to pixel, only $50 \times 50 = 2500$ pixels are needed to build a CNN model according to Fig. 10b.

$$\mathbf{X} = \begin{bmatrix} 0 & 0 + \text{rand} \\ \vdots & \vdots \\ 19 & 19 + \text{rand} \\ 20 & 20 + \text{rand} \end{bmatrix} \tag{42}$$

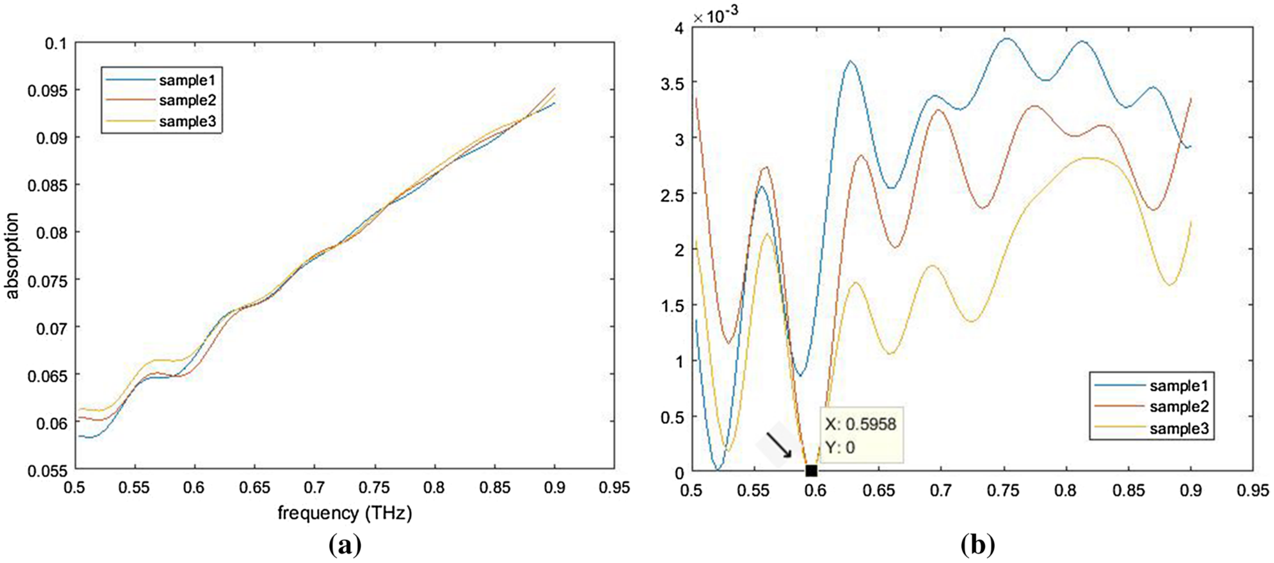

Figure 10. An NUC example based on extinction coefficient of colla corii asini: (**a**) plot of original curves (**b**) plot of processed curves. After excuting NUC, the undulation in Y-axis is reduced to only 1/9 of the original one. However, we do not assume the resultant curve as the standard spectrum. As is mentioned previously, the method is proposed for identification with CNN model at a low calculation cost.

numeric relationship	$Ry_1 = 1 + (b_2\text{-}a_1)\cdot\lvert k_1^*\rvert$	$Ry_2 = 1 + (a_2\text{-}b_1)\cdot\lvert k_2^*\rvert$	$Ry_3 = 1 + (a_2\text{-}b_1)\cdot\lvert k_3^*\rvert$	$Ry_4 = 1 + (b_2\text{-}a_1)\cdot\lvert k_4^*\rvert$	adaption
$a_1 < a_2 < b_1 < b_2$	≥1	≤1	≤1	≥1	good
$a_1 < b_1 < a_2 < b_2$	≥1	≥1	≥1	≥1	bad
$a_1 < b_1 < b_2 < a_2$	≥1	≥1	≥1	≥1	bad
$b_1 < b_2 < a_1 < a_2$	≤1	≥1	≥1	≤1	good
$b_1 < a_1 < b_2 < a_2$	≥1	≥1	≥1	≥1	bad
$b_1 < a_1 < a_2 < b_2$	≥1	≥1	≥1	≥1	bad
$a_1 = a_2 < b_1 < b_2$	≥1	≤1	≤1	≥1	good
$b_1 < a_1 = a_2 < b_2$	≥1	≥1	≥1	≥1	bad
$b_1 < b_2 < a_1 = a_2$	≤1	≥1	≥1	≤1	good
$b_1 = b_2 < a_1 < a_2$	≤1	≥1	≥1	≤1	good
$a_1 < b_1 = b_2 < a_2$	≥1	≥1	≥1	≥1	bad
$a_1 < a_2 < b_1 = b_2$	≥1	≤1	≤1	≥1	good
$a_1 = a_2 < b_1 = b_2$	≥1	≤1	≤1	≥1	good
$b_1 = b_2 < a_1 = a_2$	≤1	≥1	≥1	≤1	good

Table 1. Adaption analysis of algorithm: a case study by discussing numeric relationship between a_1, a_2, b_1 and b_2.

Discussion

It is important to discuss the adaption of NUC before using it to pretreat THz curves. We start the discussion by analyzing necessary conditions for reasoning.

All of calculations are based on the convex hull of normalized curve. Therefore, the algorithm does not adapt to cases where convex hull does not exist (all points are in a line). This is seldom found in real practice because of random error and various values at multiple frequency. Thus, we assume the proposed method works for extracting undulation information if other condition is not considered.

In addition, the searching of point in convex hull is based on comparing slope and therefore it is important to assure that the slope can be calculated regardless of the shape of curve. All of points have different x coordinate before and after normalization. Consequently, the slope can't approach infinity. Although the values at different frequency may equal, the slope can't be 0 if two different points are involved in calculation (the cardinal number exceeds 1) because that the searching scope is beyond a_1, a_2, b_1, b_2. P_1, P_2, Q_1, Q_2 belong to A_3, A_4, B_3, B_4, respectively, only if the curve goes through (0, 1), (1, 1), (0, 0), (1, 0), respectively. In other words, set A_3, A_4, B_3, B_4 contain one or more non-zero elements or contain only one element 0. In order to calculate in all cases, we expand the definition of slope in narrow sense but define piecewise function '*slope*' according to (31). It is expected that at least one of the Ry_i is minor than 1 because we want to reduce the undulation; the adaption is bad if $Ry^* \geq 1$. A detailed case study is listed in Table 1.

As is found, if both $b_2 > a_1$, $a_2 > b_1$ are satisfied, the adaption is bad because $Ry_i \geq 1$ for i $=\in \{1, 2, 3, 4\}$. When the above mentioned two expressions are neither satisfied, we would conclude $b_2 < a_1 \leq a_2 < b_1$ that conflicts with $b_1 \leq b_2$. If $b_2 < a_1$ and $a_2 > b_1$, $k_1{}^* \neq 0$ and $k_4{}^* \neq 0$ because $a_1 \neq 1$ and $b_2 \neq N$. Thus, $Ry^* < 1$; if $b_2 > a_1$ and $a_2 < b_1$, $k_2{}^* \neq 0$ and $k_3{}^* \neq 0$ because $a_2 \neq N$ and $b_2 \neq 1$. In summarize, the algorithm adapts to process curves which are governed by (43). That's the reason why a judgement is needed to check if the curve can be effectively processed by the algorithm. It turns out that after one shear transformation, the adjusted curve may be proceeded further as expression (43) is still met. Thus, one can iterate the process discussed above until expression (43) is no longer valid. The algorithm is destined to terminate after several circulations because every polygon has a dimension orthogonal to one of its edges, which is smaller than any other dimension.

$$(b_2 - a_1)(a_2 - b_1) < 0 \tag{43}$$

The computation cost of algorithm is also concerned by potential users. The time required is tightly associate with shape of curve. Given number of points, the cost to obtain A, A_1, A_2 B, B_1, B_2 can be estimated. However, the cardinal number of A_3, A_4, B_3, B_4, would be greater for curves with drastic fluctuation than smooth curves, which may add pressure to slope calculation and comparison. Unlike other common methods to pretreat THz curve, including MSC, SG filter, median filter, the time cost varies notably.

The most suitable curves are estimated curves resemble that of colla corii asini, which have overall increasing or decreasing trend, large range but small local undulation relatively. After NUC, the range is compressed significantly, which allows fewer pixels to present supplementary space if quantification of absorption change over pixel is fixed. Considering that the scattering by grains tend to cause uptrend baseline in experiments conducted by THz-TDS, featureless smooth curve of some materials may magnify their local fluctuation after NUC in finer level, make it possible to identify them with CNN models.

Conclusion

The conversion of THz spectroscopic data to 2-D image for CNN model can be achieved if a unit interval in absorption is quantified by definite number of pixels. Only a few pixels are hit by curve and the rest pixels do not provide effective information regarding curve shape. In order to reduce calculation cost of CNN, it is possible to reserve effective undulation and confine Y-range by adjusting curve in a normalized space and restore it into value-frequency space. Such operation is narrow undulation constraint (NUC), whose kernel thought is to confine curve with narrow parallel lines repeatedly and adjust range in Y by shear transformation.

A fast algorithm is proposed to achieve such goal whose kernel is searching critical points in the edge of convex hull and comparing slopes. The algorithm, described in several steps, is further illustrated and discussed from aspect of its adaption to THz curve. A number of set and definitions are purposefully built in this work, which facilitates understanding of calculations. The study suggests that studies on computer graphics also contributes to pretreatment of THz wave, which may be ignored in previous studies.

Data availability

The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Received: 21 April 2022; Accepted: 30 September 2022
Published online: 24 October 2022

References

1. Sibik, J. & Zeitler, J. A. Direct measurement of molecular mobility and crystallisation of amorphous pharmaceuticals using terahertz spectroscopy. *Adv. Drug Deliv. Rev.* **100**, 147–157 (2016).
2. Whelan, P. *et al.* Case studies of electrical characterisation of graphene by terahertz time-domain spectroscopy. *2D Mater.* **8**(2), 022003 (2021).
3. Han, P., Wang, X. K. & Zhang, Y. Time-Resolved Terahertz Spectroscopy Studies on 2D Van der Waals Materials. *Adv. Opt. Mater.* **8**(3), 1900533 (2020).
4. Bawuah, P. & Zeitler, J. A. Advances in terahertz time-domain spectroscopy of pharmaceutical solids: A review. *Trac-trends Anal. Chem.* **139**, 116272 (2021).
5. Park, H. & Son, J. H. Machine Learning Techniques for THz Imaging and Time-Domain Spectroscopy. *Sensors.* **21**(4), 1186 (2021).
6. Rao, J. *et al.* Identification of four origins of curcuma based on terahertz time-domain spectroscopy. *Laser Optoelectron. Prog.* **58**(22), 2200002 (2021).
7. Yin, X. *et al.* Exploring the complementarity of THz pulse imaging and DCE-MRIs: Toward a unified multi-channel classification and a deep learning framework. *Comput. Methods Progr. Biomed.* **137**, 87–114 (2016).
8. Afsah-Hejri, L. *et al.* Terahertz spectroscopy and imaging: A review on agricultural applications. *Comput. Electron. Agric.* **177**, 105628 (2020).
9. Afsah-Hejri, L. *et al.* A comprehensive review on food applications of terahertz spectroscopy and imaging. *Compr. Rev. Food Sci. Food Saf.* **18**(5), 1563–1621 (2019).
10. Ji, Ni. *et al.* Terahertz spectroscopic identification with diffusion maps. *Spectrosc. Spectr. Anal.* **37**(8), 2360–2364 (2017).
11. Wang, Z., Luo, J., Li, X., et al. Spectroscopy and Spectral Analysis 40(2): 391-396 (2020).
12. Ma, S. *et al.* Terahertz spectroscopic identification with deep belief network. *Spectrosc. Spectr. Anal.* **35**(12), 3325–3329 (2015).
13. Li, T. *et al.* A method of amino acid terahertz spectrum recognition based on the convolutional neural network and bidirectional gated recurrent network model. *Sci. Program.* **2021**, 2097257 (2021).
14. Sarjas, A. *et al.* automated inorganic pigment classification in plastic material using terahertz spectroscopy. *Sensors.* **21**(14), 4709 (2021).
15. Yang, S. *et al.* Determination of the geographical origin of coffee beans using terahertz spectroscopy combined with machine learning methods. *Front. Nutr.* **8**, 680627 (2021).
16. Zeng, J. *et al.* A review of the discriminant analysis methods for food quality based on near-infrared spectroscopy and pattern recognition. *Molecules* **26**(3), 749 (2021).

17. An, P. T., Huyen, P. T. T. & Le, N. T. A modified graham's convex hull algorithm for finding the connected orthogonal convex hull of a finite planar point set. *Appl. Math. Comput.* **397**, 125889 (2021).
18. Klimenko, G., Raichel, B. & Van Buskirk, G. Sparse convex hull coverage. *Comput. Geom. Theory Appl.* **98**, 101787 (2021).
19. Zeng, M. *et al.* Maximum margin classification based on flexible convex hulls. *Neurocomputing* **149**, 957–965 (2015).
20. Jarvis, R. A. On the identification of the convex hull of a finite set of points in the plane. *Inf. Process. Lett.* **2**, 18–21 (1973).
21. Graham, R. An efficient algorithm for determining the convex hull of a finite planar set. *Inf. Process. Lett.* **26**, 132–133 (1972).
22. Barber, C. B., Dobkin, D. P. & Huhdanpaa, H. The quickhull algorithm for convex hulls. *ACM Trans. Math. Softw.* **22**(4), 469–483 (1996).
23. Preparata, F. P. & Hong, S. J. Convex hulls of finite sets of points in two and three dimensions. *Commun. ACM.* **20**(2), 87–93 (1977).
24. Andrew, A. M. Another efficient algorithm for convex hulls in two dimensions. *Inf. Process. Lett.* **9**(5), 216–219 (1979).

Acknowledgements

This work is supported by Major Scientific and Technological Innovation Projects in Shandong Province (2019JZZY010448), Science Foundation of Shandong Academy of Sciences for Youth (2020QN0030), Natural Science Foundation of Shandong Province (ZR2020QF089,ZR2020KF007) and National Nature Science Foundation (81830111).

Author contributions

Y.L.: Conceptualization, methodology, writing, software. L.L.: formal analysis, software. K.L.: formal analysis, review; methodology. Z.W.: validation, project administration. T.C.: validation. W.X.: funding acquisition.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to Z.W.

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.