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. The curve of absorption or extinction coefficient is so related to the constituents of material that pattern recognition is conducted in various background. 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. 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. 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). Enlightened by the basic thought, the proposed method is named narrow undulation constraint (NUC).
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 \( X_{N \times 2}, T_{N \times 2}, Z_{2 \times 2} \) 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 \( L_{N \times 2} \), the matrix to denote cropping line.

\[
X = \begin{bmatrix} f_1 & \text{value}(f_1) \\ \vdots & \vdots \\ f_N & \text{value}(f_N) \end{bmatrix} \quad (1)
\]

\[
T = \begin{bmatrix} -f_1 - \min(\text{value}) \\ \vdots \\ -f_1 - \min(\text{value}) \end{bmatrix} \quad (2)
\]

\[
Z = \begin{bmatrix} \frac{1}{\max(\text{value}) - \min(\text{value})} \\ 0 \\ \frac{1}{\max(\text{value}) - \min(\text{value})} \end{bmatrix} \quad (3)
\]

\[
Y = (X + T)Z - L)Z^{-1} - T \quad (4)
\]

**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 \( \theta \), 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; \( A_1 \subset A \) and \( B_1 \subset B \). For convenience, the element in set A and B are denoted by \( a \) and \( b \) respectively.

\[
(f_i, \text{value}(f_i)) \text{ is a local minimum, s.t. } \begin{cases} \text{value}(f_i) < \text{value}(f_{i-1}) \\ \text{value}(f_i) < \text{value}(f_{i+1}) \end{cases}, 2 \leq i \leq N - 1 \quad (5)
\]

\[
(f_i, \text{value}(f_i)) \text{ is a local maximum, s.t. } \begin{cases} \text{value}(f_i) > \text{value}(f_{i-1}) \\ \text{value}(f_i) > \text{value}(f_{i+1}) \end{cases}, 2 \leq i \leq N - 1 \quad (6)
\]

\[
(f_i, \text{value}(f_i)) \text{ is a bump point, s.t. } \begin{cases} (\text{value}(f_i) - \text{value}(f_{i-1})) \cdot (\text{value}(f_{i+1}) - \text{value}(f_i)) > 0 \\ \text{value}(f_{i+1}) - \text{value}(f_i) < \text{value}(f_i) - \text{value}(f_{i-1}) \end{cases} \quad (7)
\]

\[
(f_i, \text{value}(f_i)) \text{ is a pit point, s.t. } \begin{cases} (\text{value}(f_i) - \text{value}(f_{i-1})) \cdot (\text{value}(f_{i+1}) - \text{value}(f_i)) > 0 \\ \text{value}(f_{i+1}) - \text{value}(f_i) > \text{value}(f_i) - \text{value}(f_{i-1}) \end{cases} \quad (8)
\]

\[
A = \{i | \text{value}(f_i) \geq \text{value}(f_j), 1 \leq i, j \leq N, i \text{ and } j \text{ are integers} \} \quad (9)
\]
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 $Q_2$ 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).

\[
\begin{align*}
A_1 &= \{i \mid (f_i, \text{value}(f_i)) \text{ is a local maximum}\} \quad (10) \\
A_2 &= \{i \mid (f_i, \text{value}(f_i)) \text{ is a bump}\} \quad (11) \\
B &= \{i \mid \text{value}(f_i) \leq \text{value}(f_i), 1 \leq i, j \leq N, i \text{ and } j \text{ are integers}\} \quad (12) \\
B_1 &= \{i \mid (f_i, \text{value}(f_i)) \text{ is a local minimum}\} \quad (13) \\
B_2 &= \{i \mid (f_i, \text{value}(f_i)) \text{ is a pit}\} \quad (14)
\end{align*}
\]

**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 $Q_2$ 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).

\[
\begin{align*}
a_1 &= \text{min}(A) \quad (15) \\
a_2 &= \text{max}(A) \quad (16) \\
b_1 &= \text{min}(B) \quad (17) \\
b_2 &= \text{max}(B) \quad (18)
\end{align*}
\]

**Step 3:** Check if curve can be processed effectively by the algorithm.
Check if \((a_i - b_j)(b_i - a_j) < 0\). If this condition is met, the curve can be processed effectively. Just follow Step 4–8 and \(Y = X\), then restart from Step 1. If \((a_i - b_j)(b_i - a_j) > 0\), the current \(X\) 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 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.

\[
X_{\text{normalized}} = (X + T)Z
\]

\[
x(i) = \frac{f_i - f_1}{f_N - f_1} \leq i \leq N, i \text{ is an integer}
\]

\[
y(i) = \frac{\text{value}(f_i) - \text{value}(f_b)}{\text{value}(f_b) - \text{value}(f_a)} \leq i \leq N, i \text{ is an integer, } a \in A, b \in B
\]

Obviously, according to our definition, we get \(y(a_1) = y(a_3) = 1\) 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 \(1 \leq i, j \leq N, i \neq j\) are integers.

\[
A_3 = \{1\} \cup \{i| i \in (A_1 \cup A_2), i < a_1\} \quad (22)
\]

\[
A_4 = \{N\} \cup \{i| i \in (A_1 \cup A_2), i > a_2\} \quad (23)
\]

\[
B_3 = \{1\} \cup \{i| i \in (B_1 \cup B_2), i < b_1\} \quad (24)
\]

\[
B_4 = \{N\} \cup \{i| i \in (B_1 \cup B_2), i > b_2\} \quad (25)
\]

\[
S = S_1 \cup S_2 \cup S_3 \cup S_4 \quad (26)
\]

\[
S_1 = \{s| s = |\text{slope}(i, a_1)|, i \in A_3\} \quad (27)
\]

\[
S_2 = \{s| s = |\text{slope}(i, a_2)|, i \in A_4\} \quad (28)
\]

\[
S_3 = \{s| s = |\text{slope}(i, b_1)|, i \in B_3\} \quad (29)
\]

\[
S_4 = \{s| s = |\text{slope}(i, b_2)|, i \in B_4\} \quad (30)
\]

\[
slope(i, j) = \begin{cases} \frac{y(i) - y(j)}{x(i) - x(j)} & i \neq j \\ 0, i = j & \end{cases} \quad (31)
\]

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.

\[
c_1 = \min(C_1), \text{s.t.} C_1 = \left\{i| i \in A_3, |\text{slope}(i, a_1)| = \min(S_1)\right\} \quad (32)
\]

\[
c_2 = \max(C_2), \text{s.t.} C_2 = \left\{i| i \in A_4, |\text{slope}(i, a_2)| = \min(S_2)\right\} \quad (33)
\]

\[
c_3 = \min(C_3), \text{s.t.} C_3 = \left\{i| i \in B_3, |\text{slope}(i, b_1)| = \min(S_3)\right\} \quad (34)
\]

\[
c_4 = \max(C_4), \text{s.t.} C_4 = \left\{i| i \in B_4, |\text{slope}(i, b_2)| = \min(S_4)\right\} \quad (35)
\]
The coordinate of \( P_1, P_2, Q_1, Q_2 \) are confirmed depending on the rotation angle \( \theta \) and rotation center. Assume parallel lines are named line 1 and line 2 according to Y offset (\( \text{offset2} > \text{offset1} \)). A universal rule to claim \( P_1, P_2, Q_1, Q_2 \) as described as follow:

\( P \) denotes point (the fixed rotation center or the critical point to be found) in line 1 and Q 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. \((x(a_i), y(a_i))\) and \((x(b_i), y(b_i))\) are fixed as rotation center of line 2 and line 1 respectively in both case 1 and case 2. As a contrast, \((x(a_i), y(a_i))\) and \((x(b_i), y(b_i))\) 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 \((x(a_i), y(a_i))\) is fixed as rotation center, \( P_1 \) is defined as \( P_1{(x(c_i), y(c_i))} \) and \( P_2 \) is defined as \( P_2{(x(a_i), y(a_i))} \). Meanwhile, \( Q_2 \) is defined as \( Q_2{(x(b_j), y(b_j))} \) and \( Q_1 \) overlaps with \( Q_2 \).

**Case 2** If point \((x(a_i), y(a_i))\) is fixed as rotation center, \( P_1 \) is defined as \( P_1{(x(a_i), y(a_j))} \) and \( P_2 \) is defined as \( P_2{(x(c_j), y(c_j))} \). Meanwhile, \( Q_1 \) is defined as \( Q_1{(x(b_i), y(b_i))} \) and \( Q_2 \) overlaps with \( Q_1 \).

**Case 3** If point \((x(b_j), y(b_j))\) is fixed as rotation center, \( P_1 \) is defined as \( P_1{(x(a_i), y(a_j))} \) and \( P_2 \) overlaps with \( P_2 \). Meanwhile, \( Q_1 \) is defined as \( Q_1{(x(c_i), y(c_j))} \) and \( Q_2 \) is defined as \( Q_2{(x(b_i), y(b_j))} \).

**Case 4** If point \((x(b_j), y(b_j))\) is fixed as rotation center, \( P_1 \) is defined as \( P_1{(x(a_i), y(a_j))} \) and \( P_2 \) overlaps with \( P_1 \). Meanwhile, \( Q_1 \) is defined as \( Q_1{(x(b_i), y(b_j))} \) and \( Q_2 \) is defined as \( Q_2{(x(c_i), y(c_j))} \).

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

\[
k^*_i = \begin{cases} 
\text{slope}(c_1, a_1), & i = 1 \\
\text{slope}(c_2, a_2), & i = 2 \\
\text{slope}(c_3, b_1), & i = 3 \\
\text{slope}(c_4, b_2), & i = 4 
\end{cases}
\] (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 \( R_y^* \) is found in (38). Correspondingly, \( k^* \) is determined according to (39). After comparing \( R_y^* \), only one case is reserved as the final result and the corresponding rotation centers are recorded.

\[
R_{yi} = \begin{cases} 
1 + (b_2 - a_1) \times \min\{|k^*_1|, |k^*_2|\}, & i = 1 or 4 \\
1 + (a_2 - b_1) \times \min\{|k^*_2|, |k^*_3|\}, & i = 2 or 3 
\end{cases}
\] (37)

\[
R_y^* = \min\{R_{y1}, R_{y2}, R_{y3}, R_{y4}\}
\] (38)

\[
k^* = k^*_i \text{s.t. } R_y^* = R_{yi}, R_y^* = R_{yj}, \quad i, j \in \{1, 2, 3, 4\}, \quad i < j
\] (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).

\[
\text{offset}_1 = \begin{bmatrix} -b_1k^* & k^* \\
-b_2k^* & k^* \\
-b_3k^* & k^* \\
-b_4k^* & k^* 
\end{bmatrix}
\] (40)

\[
L = \begin{bmatrix} 0 & k^*x_1 + \text{offset}_1 \\
\vdots & \vdots \\
0 & k^*x_N + \text{offset}_1 
\end{bmatrix}
\] (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\textsuperscript{20}, Graham scan\textsuperscript{11}, quick hull\textsuperscript{22}, divide and
conquer23 and monotone chain24. 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 Algorithm24, 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 $Y$ value 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...
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 lie beyond the narrow band). As the processed undulation change monotonically with $\theta$, 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 $(x(a_1), 1)$, $(x(a_2), 1)$, $(x(b_1), 0)$ and $(x(b_2), 0)$ for 4 cases.

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 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 $\theta$ 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 $\theta$, 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 $(x(a_1), 1)$, $(x(a_2), 1)$, $(x(b_1), 0)$ and $(x(b_2), 0)$ for 4 cases.
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
A chemical compound is very complex. A number of researchers focus on identifying mixtures without notable features in the THz band. NUC makes it possible to extract undulations for identification while removing the overall linear trend that contributes to large range in absorption, therefore reducing the number of pixels required for the CNN model. An estimation can be made that the original curve with 50 frequency samplings and an absorption range of 0.045 needs 50 × 450 = 22,500 pixels according to Fig. 10a if the absorption range of 0.01 is quantified using 100 pixels. However, with the same ratio of value to pixel, only 50 × 50 = 2,500 pixels are needed to build a CNN model according to Fig. 10b.

\[
X = \begin{bmatrix}
0 & 0 + \text{rand} \\
\vdots & \vdots \\
19 & 19 + \text{rand} \\
20 & 20 + \text{rand}
\end{bmatrix}
\]  (42)
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 \( a_1, a_2, b_1, b_2 \), \( \{ a_3, a_4, b_3, b_4 \} \). \( P_1, P_2, Q_1, Q_2 \) 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 \( R_{ij} \) is minor than 1 because we want to reduce the undulation; the adaption is bad if \( R_{ij} \geq 1 \). A detailed case study is listed in Table 1.

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

| numeric relationship | \( R_{ij} = 1 + (b_2-a_2)\cdot|k_j^*| \) | \( R_{ij} = 1 + (a_2-b_1)\cdot|k_j^*| \) | \( R_{ij} = 1 + (a_2-a_1)\cdot|k_j^*| \) | \( R_{ij} = 1 + (b_2-a_1)\cdot|k_j^*| \) | adaption |
|----------------------|---------------------------------|---------------------------------|---------------------------------|---------------------------------|---------|
| \( a_1 < a_2 < b_1 < b_2 \) | \( \geq 1 \) | \( \leq 1 \) | \( \leq 1 \) | \( \geq 1 \) | good |
| \( a_1 < b_1 < a_2 < b_2 \) | \( \geq 1 \) | \( \geq 1 \) | \( \geq 1 \) | \( \geq 1 \) | bad |
| \( a_1 < b_1 < b_2 < a_2 \) | \( \geq 1 \) | \( \geq 1 \) | \( \geq 1 \) | \( \geq 1 \) | bad |
| \( b_1 < b_2 < a_1 < a_2 \) | \( \leq 1 \) | \( \geq 1 \) | \( \geq 1 \) | \( \leq 1 \) | good |
| \( b_1 < a_1 < b_2 < a_2 \) | \( \leq 1 \) | \( \leq 1 \) | \( \leq 1 \) | \( \leq 1 \) | bad |
| \( b_1 = a_2 = b_1 < b_2 \) | \( \leq 1 \) | \( \leq 1 \) | \( \leq 1 \) | \( \leq 1 \) | bad |
| \( b_1 = a_2 < a_1 < a_2 \) | \( \leq 1 \) | \( \leq 1 \) | \( \leq 1 \) | \( \leq 1 \) | good |
| \( b_1 = b_2 < a_1 = a_2 \) | \( \leq 1 \) | \( \leq 1 \) | \( \leq 1 \) | \( \leq 1 \) | good |
| \( a_1 = b_1 < a_2 < b_2 \) | \( \leq 1 \) | \( \leq 1 \) | \( \leq 1 \) | \( \leq 1 \) | good |
| \( a_1 = a_2 < b_1 = b_2 \) | \( \leq 1 \) | \( \leq 1 \) | \( \leq 1 \) | \( \leq 1 \) | good |
| \( a_1 < a_2 < b_1 = b_2 \) | \( \geq 1 \) | \( \leq 1 \) | \( \leq 1 \) | \( \geq 1 \) | bad |
| \( a_1 < a_2 < b_1 = b_2 \) | \( \geq 1 \) | \( \leq 1 \) | \( \leq 1 \) | \( \geq 1 \) | bad |
| \( b_1 < a_1 = a_2 < b_2 \) | \( \leq 1 \) | \( \leq 1 \) | \( \leq 1 \) | \( \leq 1 \) | good |
| \( b_1 = b_2 < a_1 = a_2 \) | \( \leq 1 \) | \( \leq 1 \) | \( \leq 1 \) | \( \leq 1 \) | good |

Figure 10. An NUC example based on extinction coefficient of colla corii asini: (a) plot of original curves (b) plot of processed curves. After executing 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.
As is found, if both $b_2 > a_i$, $a_i > b_1$ are satisfied, the adaption is bad because $R_{ij} \geq 1$ for $i = \{1, 2, 3, 4\}$. When the above mentioned two expressions are neither satisfied, we would conclude $b_i < a_i < b_i$ that conflicts with $b_i \leq b_i$. If $b_2 < a_i$ and $a_i > b_1$, $k_i^* \neq 0$ and $k_i^* \neq a$ because $a_i \neq b_2$ and $b_1$. Thus, $R_{ij} < 1$; if $b_1 > a_i$ and $a_i < b_1$, $k_i^* \neq 0$ and $k_i^* \neq a_i$, which cause $a_i \neq N$ and $b_i \neq 1$. 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.

\[(b_2 - a_1)(a_2 - b_1) < 0 \quad (43)\]

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 $A$, $A_i$, $B_i$, $B_i$, $B_i$, $B_i$ can be estimated. However, the cardinal number of $A$, $A_i$, $B_i$, $B_i$, $B_i$, $B_i$ 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 callo 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 uprend 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. *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.* **183**, 87–114 (2021).

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. Li, N. 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 (2021).

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, 123889 (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.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

© The Author(s) 2022