Real-Time Hyperbola Recognition and Fitting in GPR Data

Qingxu Dou, Lijun Wei, Derek R. Magee, and Anthony G. Cohn

***Abstract*—The problem of automatically recognizing and fitting hyperbolae from ground-penetrating radar (GPR) images is addressed, and a novel technique computationally suitable for real-time on-site application is proposed. After preprocessing of the input GPR images, a novel thresholding method is applied to separate the regions of interest from background. A novel column-connection clustering (C3) algorithm is then applied to separate the regions of interest from each other. Subsequently, a machine learnt model is applied to identify hyperbolic signatures from outputs of the C3 algorithm, and a hyperbola is fitted to each such signature with an orthogonal-distance hyperbola fitting algorithm. The novel clustering algorithm C3 is a central component of the proposed system, which enables the identification of hyperbolic signatures and hyperbola fitting. Only two features are used in the machine learning algorithm, which is easy to train using a small set of training data. An orthogonal-distance hyperbola fitting algorithm for "south-opening" hyperbolae is introduced in this work, which is more robust and accurate than algebraic hyperbola fitting algorithms. The proposed method can successfully recognize and fit hyperbolic signatures with intersections with others, hyperbolic signatures with distortions, and incomplete hyperbolic signatures with one leg fully or largely missed. As an additional novel contribution, formulas to compute an initial "south-opening" hyperbola directly from a set of given points are derived, which make the system more efficient. The parameters obtained by fitting hyperbolae to hyperbolic signatures are very important features; they can be used to estimate the location and size of the related target objects and the average propagation velocity of the electromagnetic wave in the medium. The effectiveness of the proposed system is tested on both synthetic and real GPR data.**

***Index Terms*—Buried asset detection, column-connection clustering (C3) algorithm, ground-penetrating radar (GPR), hyperbola recognition, machine learning, orthogonal-distance fitting.**

I. Introduction

As a NONDESTRUCTIVE tool for the investigation of shallow subsurface, ground-penetrating radar (GPR) has been widely used in the detection and mapping of subsurface utilities such as pipes and cables [1]. There are typically two pattern shapes in the B-scan images of GPR: hyperbolic curves and linear segments [2]. Hyperbolic curves are due to objects with a cross-section size on the order of the radar pulse wavelength; linear segments stem from planar interfaces between layers with different electrical impedance values. Because of system noise, the heterogeneity of the medium, and mutual wave interactions, GPR images are usually noisy. It is a complex task to automatically extract hyperbolae from GPR data. Considerable research has been devoted in this area, and many different strategies have been employed to tackle this topic, e.g., [3]–[11]. In addition, if the parameters of a hyperbolic signature can be obtained by fitting a hyperbola to it, the parameters can be used to estimate the location and size of the related target object and the average propagation velocity of the electromagnetic wave in the medium [12].

In [9] and [13]–[15], the generalized Hough transform is used to find the parameters of hyperbolae. It is time consuming to determine the parameters of a hyperbola with generalized Hough transform algorithms because the algorithms need to be performed in a space with at least 4 dimensions. In addition, the accuracy of a generalized Hough transform algorithm depends on the discretization of the parameters. Increasing the discretization of the parameters moderately could lead to dramatically increasing the computing time. In [16], the generalized Hough transform method was extended to record the associative sets of position/time data pairs that form a contribution to each bin in the Hough accumulator space, which can then be used with a conventional least squares algorithm to reveal the object position, depth, and radius or velocity. In [17], the edges in the GPR images are detected first and then followed by an edge-fitting algorithm. This algorithm is only suitable for very clean GPR images. Otherwise, it would be very difficult to group the points detected from a certain edge for fitting. In [10], an edge detector is also applied to detect edges from GPR images. Although this method can be applied on complex GPR images, in fact, no fitting is directly applied on the detected edge points; hence, only the apexes of the hyperbolae are detected, and other parameters of the related hyperbolae are missed, which are essential for identifying other properties of the utilities such as the size of the utilities [12] and even the materials of the utilities [11].

Another type of approach uses machine learning methods to narrow down the regions including hyperbolae in the first step, and then, a fitting method is applied to find the hyperbola parameters [9], [18]. In [18], after the regions including hyperbolae are extracted with a neural network, an edge detector is employed to detect edges in the extracted regions, and then, the parameters of hyperbolae are extracted through a generalized Hough transform. In [9], the Viola–Jones algorithm [19] is employed to extract the regions believed to contain hyperbolae, followed by a generalized Hough transform fitting based on the

Manuscript received February 15, 2016; revised June 28, 2016; accepted July 5, 2016. Date of publication September 21, 2016; date of current version November 29, 2016. This work was supported in part by the Engineering and Physical Sciences Research Council under Grant EP/F06585X/1 and Grant EP/K021699/1 and in part by the European Commission under FP7 Project NeTTUN, Grant Agreement 280712. (*Corresponding author: Qingxu Dou.*)

The authors are with the School of Computing, University of Leeds, Leeds LS2 9JT, U.K. (e-mail: Q.Dou@leeds.ac.uk; L.J.Wei@leeds.ac.uk; D.R.Magee@leeds.ac.uk; A.G.Cohn@leeds.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2016.2592679

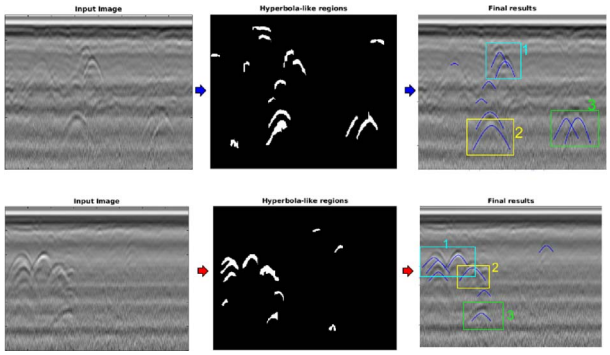


Fig. 1. Examples of difficult scenarios that can be tackled by the proposed hyperbola recognition and fitting method. The first column contains the input GPR images, the second column contains the candidate hyperbolic signatures, and the third column contains the fitted hyperbolae with difficult scenarios, including intersecting hyperbolae in *rectangles 1* and incomplete or distorted hyperbolae in *rectangles 2* and *rectangles 3*.

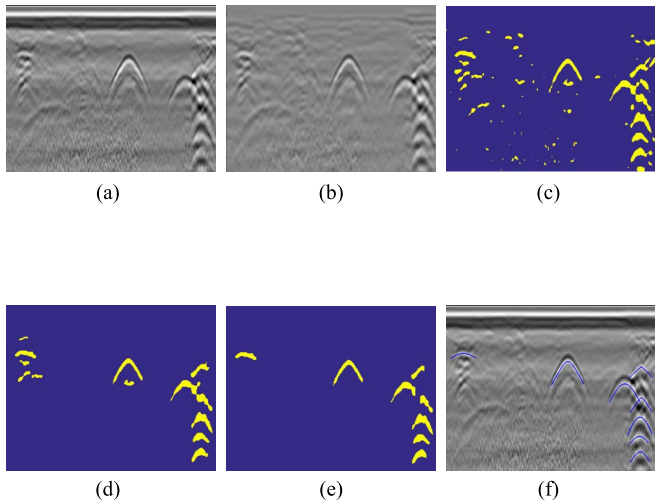


Fig. 2. Illustration of the application of the proposed technique on the bright regions (as described in Section II-A) of a real GPR image. (a) Input image. (b) Image after preprocessing. (c) Regions of interest after thresholding. (d) Clusters after applying the C3 algorithm. (e) Identified hyperbolic signatures by applying the machine learning algorithm. (f) Output image from the system with fitted hyperbolae. Intersecting—with crossing tails, connected without crossing tails. Distorted—asymmetric or incomplete (best viewed in color).

detected edge points. The disadvantages of extracting hyperbola parameters through a generalized Hough transform and edge fitting are pointed out above. In addition, as pointed out by Maas and Schmalzl in [9], the quality of detection results strongly depends on the quality and size of the available data for training. The experimental statistics are very impressive with respect to recall and precision for hyperbola detection and fitting in [7], but the algorithm is only tested with synthetic data generated with GprMax [20], and the scenarios are relatively simple such as no intersection of the hyperbolic signatures is seen in the displayed GPR images. In [21], Chen and Cohn suggested a probabilistic hyperbola mixture model based on a classification expectation–maximization algorithm to extract multiple hyperbolae from a GPR image in one go. There are at least two issues worthy of further consideration. First, compared with an orthogonal circle or ellipse fitting algorithm, orthogonal hyperbola fitting algorithms are more sensitive to the configuration of the given points. The expectation–maximization algorithm starts with a general initial partition of the given points; it is difficult to guarantee the convergence of the hyperbola fitting algorithm. Second, the computation of an orthogonal hyperbola fitting algorithm is expensive. In each step, the expectation–maximization algorithm calls the hyperbola fitting algorithm multiple times.

In this paper, we propose a method to automatically detect and fit hyperbolae to GPR images. The proposed multistage approach can deal with complex GPR images and, particularly, can recognize and fit hyperbolic signatures in some difficult scenarios as shown in Fig. 1, such as hyperbolic signatures with intersections with others, hyperbolic signatures with distortions, and incomplete hyperbolic signatures with one leg fully or largely missed, possibly due to local velocity changes. The fitted parameters of the hyperbolic signatures can then be used to estimate the location and size of the related target objects and the average propagation velocity of the signals in the medium for future applications.

The proposed system is composed of four stages; an application of the proposed system is shown in Fig. 2. First, a preprocessing procedure is applied to the input image, and then, a threshold value is automatically selected based on the results of edge detection. With this threshold value, the regions of interest are separated from the background. With the proposed column-connection clustering (C3) algorithm, the regions of interest are separated into different clusters. As previously mentioned, a hyperbola must fit a hyperbolic signature to guarantee convergence. Which regions can be regarded as a hyperbolic signature? A machine learnt model is applied to identify the hyperbolic signatures. As previously pointed out, it is necessary to fit a hyperbola to each hyperbolic signature to obtain the corresponding parameters that can be used to estimate the location and size of the target object and the signal propagation velocity in the medium [12].

The C3 algorithm is the central component of this work. The previous clustering algorithms are either based on the distance between points [22], [23] or the density of points within a certain area [24], [25]. They are not capable of separating connected regions or segmenting two hyperbola signatures with an intersection. The proposed C3 algorithm is based on matching sequences of elements in adjacent columns with the same row numbers. The output clusters of this algorithm include different combinations of connected blocks, and one block can belong to multiple different clusters. With this algorithm, most hyperbolic signatures can be segmented from other regions even if they are connected or have intersections before clustering. Without this step, the proposed machine learning algorithm and hyperbola fitting algorithm cannot be applied.

The hyperbola fitting algorithm is also a crucial component of this work. There is a large body of conic fitting algorithms in the literature [26]–[30]. Compared with algebraic distance, orthogonal distance is invariant to transformations in Euclidean space; therefore, orthogonal-distance fitting algorithms are more robust and accurate than algebraic distance fitting algorithms [26]. In this paper, we introduce a least squares orthogonal-distance fitting algorithm for "south-opening" hyperbolae based

on the work of Ahn *et al.* in [26]. The efficiency of the fitting algorithm makes this system suitable for real-time on-site application. In addition, a novel way to compute the initial hyperbola parameters directly from the given points is introduced. Compared with using algebraic hyperbola fitting results as the initial hyperbola for the orthogonal hyperbola fitting as in [26], [29], and [31], the initial hyperbola computed with the proposed method is usually closer to the final fitted hyperbola; this makes the fitting algorithm even more efficient.

The rest of this paper is organized as follows. We present the proposed C3 algorithm and the related GPR image preprocessing schemes in Section II, which is followed by a description of the machine learning algorithm for hyperbolic signature identification in Section III. The orthogonal-distance hyperbola fitting algorithm and the hyperbola initialization procedure are presented in Section IV. The experimental results are shown and analyzed in Section V, and finally, conclusions are drawn in Section VI.

II. Column-Connection Clustering Algorithm

In this section, we present the proposed column-connection clustering (C3) algorithm and the related preprocessing procedures on real data.

A. Adaptive Thresholding of the Input Images

Before applying the proposed clustering algorithm, a series of processing techniques is employed on real GPR images. In Fig. 2(a), it is shown that some regions including hyperbolic signatures, a strip at the upper part of the image, and some small irregular regions have higher responses. It is a common feature for a GPR image to have a bright strip at the top of the image, which is due to the reflectance of the ground surface. In the preprocessing, a moving average filter is applied to the input image to reduce the noise, and then, the ensemble mean of each row is subtracted to eliminate the bright ground surface reflectance strip. An example image after the preprocessing is shown in Fig. 2(b). The window size of the filter should not be too large but within a certain range; the experimental results are not that sensitive to it. In our experiments, we tried with 3×3, 5×5, and 7×7 (in pixels), and very similar results were obtained. The experimental results shown in this paper were done with a filter window size of 3×3.

The regions corresponding to the maxima of the positive phase (bright) or minima of the negative phase (dark) of the reflected radar signal are the regions of interest for identifying hyperbolae. Since the dark regions of an image correspond to the bright regions of its inverse image, in the following sections, we shall focus on the bright regions representing high responses. If a suitable threshold value can be selected to separate the regions of interest (high responses) from the background, it simplifies further processing. To pick a threshold to separate two regions with different intensities in an image, it is natural to use the intensity value of a pixel on the boundary between these two regions as the threshold. In our work, a large number of regions of interest need to be separated from the background, and many boundaries between the regions of interest and the background are involved. We decide to pick a threshold, which relates to the average of the intensity values of the boundary points. First, an edge detector is used to extract the edges between regions of interest and the background to obtain the intensity values of the edge points. If we use the average of all the edge points as the threshold, then experiments give good recall values but bad precision. If we average by chopping off some darker edge points, the balance between the value of recall and the value of precision improves. However, if we chop off too many darker edge points before averaging, the balance worsens. Hence, only the edge pixel intensities that are greater than a certain percentage of the value of the highest edge pixel intensity are used for averaging to obtain the threshold value. The computation of the threshold can be performed with the following expression:

$$\text{threshold}_b = \text{mean}\{Ie | Ie > \rho_b \times \text{Max}_{Ie}\} \tag{1}$$

where mean is a function for computing the average among a set of values, Ie is the intensity value of an edge pixel, Max_{Ie} is the highest edge intensity value, and ρ_b is a fraction $(0 < \rho_b < 1)$.

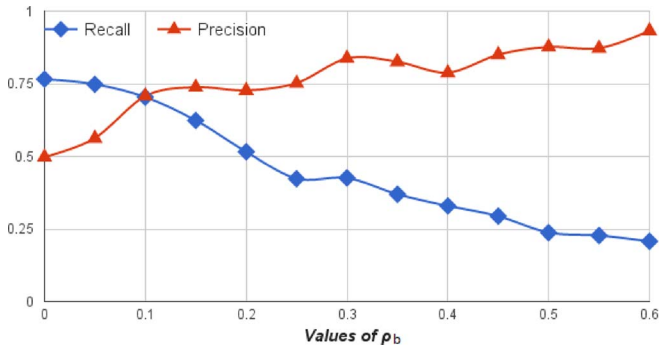


Fig. 3. Illustration of effect of ρ_b in (1) on the values of recall and precision of hyperbola fitting when applying the proposed system on a group of GPR images. Recall $= tp/(tp + fn)$, precision $= tp/(tp + fp)$, where tp is the number of correctly fitted hyperbolae by the algorithm; fn is the number of hyperbolae in the ground truth, which are not correctly fitted by the proposed algorithm; and fp is the number of fitted hyperbolae not included in the ground truth.

The effect of the value ρ_b on the values of recall and precision of fitted hyperbolae when applying the proposed system on a group of real GPR images is demonstrated in Fig. 3. It can be seen that with the value of ρ_b increasing, recall decreases while precision increases. Balancing between these two factors, 0.1 is used in our experiments.

The proposed adaptive thresholding algorithm is also compared with other existing thresholding methods in the literature, which are totally different from each other: the statistical thresholding method in [32], the maximum entropy thresholding method in [33], and the unimodal thresholding method in [34]. The method proposed by Kapur *et al.* in [33] was also used by Pasolli *et al.* in [11] on GPR images to separate the hyperbola regions from the background. As shown in Fig. 4(c), it seems that the threshold given by the statistical thresholding method in [32] is too low, and it only removes those dark areas; the threshold given by the maximum entropy thresholding method in [33] is usually too high to retain all the hyperbola regions [see Fig. 4(d)]. The outputs from the unimodal thresholding

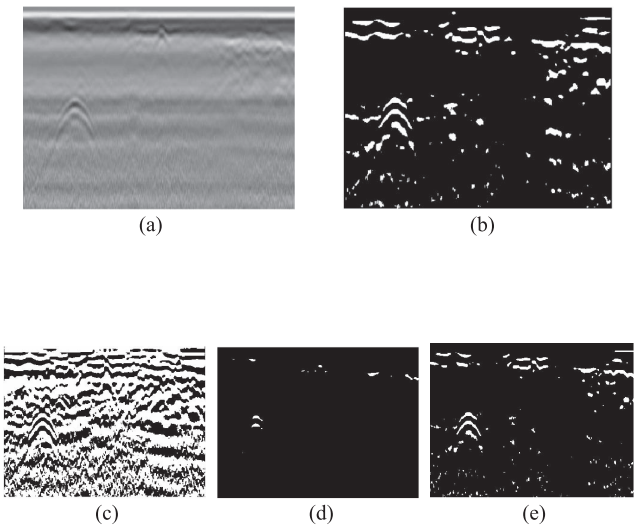


Fig. 4. Comparison of different thresholding methods for bright regions on one real GPR image. (a) Original image. (b) Proposed thresholding. (c) Statistical thresholding [32]. (d) Maximum entropy thresholding [33]. (e) Unimodal thresholding [34].

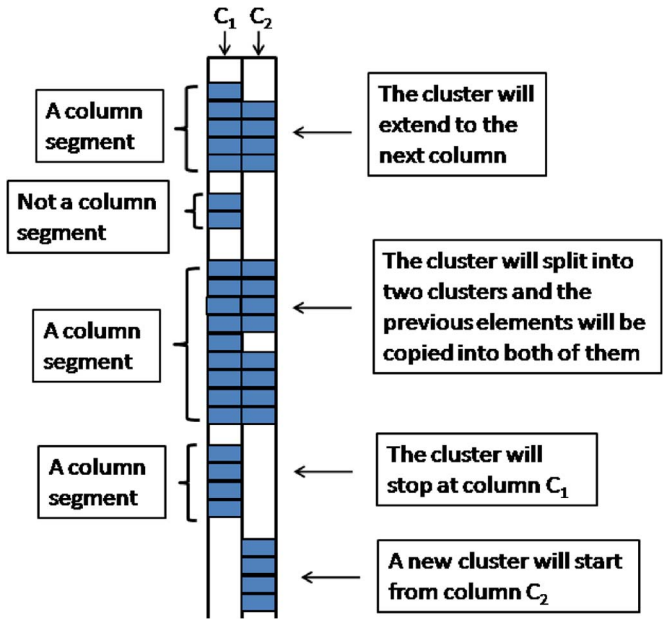


Fig. 5. Illustration of the C3 algorithm (see detailed explanation in the text).

method [34] [see Fig. 4(e)] are very similar to those of the proposed method [see Fig. 4(b)], although based on totally different computation strategies. Further comparison on these two methods with detailed statistics can be found in Section V.

With the computed threshold value, the original image is converted into a binary image [e.g., Fig. 4(b)], which is used for further processing. In a binary image, if the value of a pixel is nonzero, it is regarded as a point. On the other hand, if the value of a pixel is zero, it is regarded as background.

B. Column-Connection Clustering Algorithm

After an input image is converted into a binary image by thresholding, the C3 algorithm is applied to separate the selected regions into different clusters. To explain the C3 algorithm clearly, two concepts should be clarified first: *Column Segment* and *Connecting Elements* of two column segments from adjacent columns.

Column Segment: When searching along a column of a binary image, if the number of consecutive points is equal to or higher than a predefined number s, then this group is called a column segment. For example, in Fig. 5, if the value of s is defined as 4, then there are three column segments along column C_1. The second group is not a column segment as there are only two consecutive elements in this group. The purpose of selecting a threshold s for column segments is for noise resistance. The criterion for choosing it depends on the noise level of the sensor, the radar central frequency, and the sampling frequency. Concretely, the maximum value of s is proportional to the sampling frequency f_s and inversely proportional to the radar frequency f_c. An ideal value of s should be bigger than most of the noise but lower than $k \cdot f_s/f_c$ (k is a constant) so as to reject most of the noise and remain the signal. In our experiments, the value of s is the same for different parts of the image.

Connecting Elements: The location of a point in a certain column segment is defined by its row number. If we say two adjacent column segments have connecting elements, it means they have elements from the same row. In this paper, we only compare the elements between two column segments that are from adjacent columns. For example, in Fig. 5, there are four connecting elements between the first column segments from column C_1 and column C_2.

In Fig. 5, if column C_1 is the first column scanned, then after searching along this column, the seeds of three clusters are generated. We call them clusters 1–3 from top to bottom. Next, column C_2 is scanned, and the column segments from this column are obtained. The first column segment from column C_2 has four connecting elements with the first column segment from column C_1. If two column segments from adjacent columns have connecting elements, then the cluster extends to the next column to include the elements of the column segment from the later column. Thus, cluster 1 is extended to column C_2. There are two column segments of column C_2, which have four connecting elements with cluster 2 of column C_1. In this situation, cluster 2 extends to column C_2 and splits into two clusters 2_a and 2_b. All the elements in cluster 2 are associated with both clusters with the elements in the second column segment of column C_2 added to cluster 2_a and the elements in the third column segment of column C_2 added to cluster 2_b. As for the third column segment in column C_1, since there is no connecting element in column C_2 with it, cluster 3 stops at column C_1. The fourth column segment in column C_2 has four elements, which is no less than s, and there is no connecting element in the previous column C_1; therefore, a new cluster starts from column C_2 with the elements in the fourth column segment as the seeds.

This procedure is performed until the last column is scanned to obtain all the clusters based on column connection. This algorithm is symmetric with respect to the scanning direction, i.e., there is no difference in performing the scanning procedure

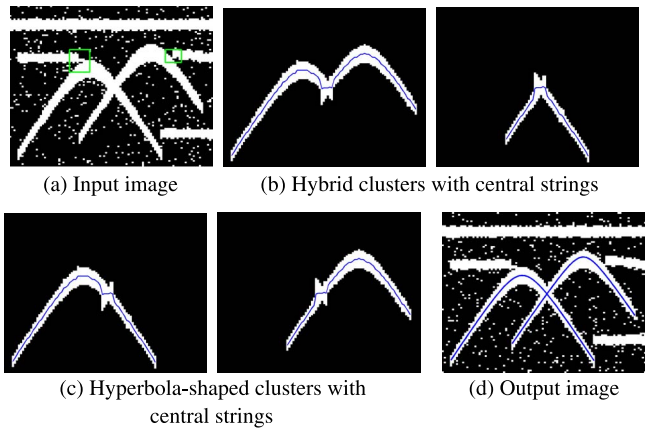


Fig. 6. Application of the proposed system on a synthetic data set. Some output clusters of the C3 algorithm with the central strings are displayed in (b) and (c). The fitted hyperbolae are shown in (d).

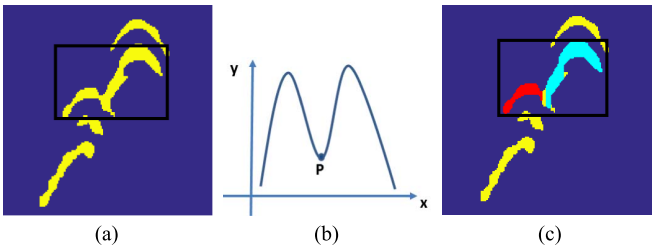


Fig. 7. Further segmentation on connected hyperbola signatures (best viewed in color). (a) After one step of C3. (b) Schematic of two connected hyperbolae. (c) After further segmentation.

from left to right or from right to left. The outputs of the clustering algorithm with one GPR image are shown in Fig. 6(b) and (c). For each output cluster from the C3 algorithm, a *central string*, which is the curve connecting the middle points of the elements in each column, is computed as shown in Fig. 6(b). The *central string* is a very important feature in the C3 algorithm; it is used for further segmentation, machine learning, and hyperbola fitting. Physically, the calculated central string corresponds to the peak point of the reflected signal.

The C3 clustering algorithm can separate hyperbolic signatures with intersections. In Fig. 6, two hyperbolic signatures, which intersect each other, are separated by the C3 algorithm as displayed in Fig. 6(c). This example is based on synthetic data. In real GPR images, the intersections between two hyperbolic signatures are more complicated. In some cases, due to the low strength of response, the parts below the intersection point are missed as the cluster shown within the rectangle window in Fig. 7(a). In this situation, the C3 algorithm described so far cannot separate them from each other. The whole region is usually identified as a nonhyperbolic signature in the machine learning step, and two hyperbolae are missed. To deal with this situation, the aforementioned C3 algorithm is extended with a further segmentation step.

Suppose the curve shown in Fig. 7(b) is the central string of an output cluster from the first step of the C3 algorithm. It is similar to the situation where two hyperbolae intersect each other at point P and the parts below point P are not detected. Mathematically, the first derivative at point P is 0, and the second derivative at point P is positive; point P is detected by checking its first and second derivatives, and the related cluster is broken at the column corresponding to point P. In the final output of C3 fed to the machine learning algorithm (which will filter out nonhyperbolic shaped responses), the original cluster before this step is also included for avoiding misjudgements.

The C3 algorithm is also helpful for eliminating image noise. In Fig. 11, it is shown that the original image is noisy but the images of the separated clusters are clean. This is achieved with the help of predefined parameter s. For a reasonable value of s such as 2 or 3, the number of the consecutive noise points along a column is usually not as high as s. Therefore, almost all the noisy points are eliminated in the clustering step. By tuning the value of s, the proposed algorithm can deal with images with different noise levels. We tried different values for parameter s in our experiments. The best results were obtained using $s = 3$. Hence, we set s equal to 3 in the shown experimental results.

The pseudocode of the proposed clustering algorithm can be presented as follows:

```
for i from min_column to max_column do
    if i == min_column
       for j from 1 to num_col_seg_c do
          cell{j,1} = col_seg_c(j);
       end
    else
       for j from 1 to num_col_seg_c do
           record = zeros(1,num_col_seg_p);
          for k = 1 to num_col_seg_p do
             n=num_same_elements(col_seg_
             c(j),
             col_seg_p(k));
           if n >= s && record(j) == 0
              cell{j,1} = [cell{j,1} col_seg_
              c(j)];
              record(j) = 1;
           elseif n >= s && record(j) == 1
              kk = size(cell,1) + 1;
              cell{kk,1} = cell{j};
              cell{kk,1}=[cell{kk,1} col_seg_
              c(j)];
           elseif n < s
              kk = size(cell,1) + 1;
              cell{kk,1} = col_seg_c(j);
           end
         end
       end
    end
end
```

For a cluster containing one hyperbolic signature, a hyperbola is fitted to this cluster to obtain its parameters. Which output clusters should be regarded as a hyperbolic signature? We answer this question by a machine learnt model for identifying hyperbolic signatures, which is explained in the next section.

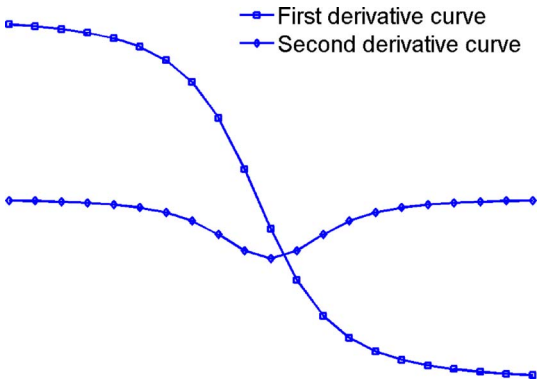


Fig. 8. First and second derivative curves of a south-opening hyperbola on a domain symmetric to the hyperbola center. The marker points on each curve make up a template.

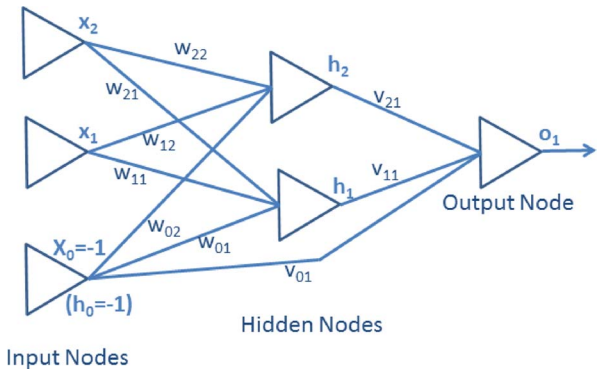


Fig. 9. Neural network diagram.

III. Machine Learning Algorithm for Identifying Hyperbolic Signatures

In this section, we present a machine learning method for identifying hyperbolic signatures.

A. Feature Extraction for a Neural Network Classification Algorithm

To successfully identify hyperbolic signatures from the outputs of the C3 algorithm, it is necessary to extract attributes that characterize hyperbolic signatures and distinguish them from other undesired clusters and composite clusters of more than one hyperbola.

In a GPR image, the detected hyperbolae are manifested as "south-opening" branches. The general equation of a "south-opening" branch of a hyperbola is written as

$$\frac{(y-y_0)^2}{a^2}-\frac{(x-x_0)^2}{b^2}=1, \quad \text{with} \quad y<y_0 \tag{2}$$

where y and x relate to the values along the vertical and horizontal axes, the vertical axis y is proportional to the two-way travel time of waves, and the horizontal axis x is the distance along the measured direction. (x_0, y_0) is the center of the hyperbola, a is the length of the semi-major axis, and b is the length of semi-minor axis.

The first and second derivatives of the function expressed by (2) have the following form:

$$\frac{dy}{dx}=-\frac{a}{b}\frac{x-x_0}{\sqrt{(x-x_0)^2+b^2}} \tag{3}$$

$$\frac{d^2y}{dx^2}=-\frac{ab}{((x-x_0)^2+b^2)^{\frac{3}{2}}}. \tag{4}$$

The graphs of the functions expressed by (3) and (4) on a domain symmetric to the center of the hyperbola are presented in Fig. 8. It can be seen that, on a domain symmetric to the center of a hyperbola, the first derivative and the second derivative of a "south-opening" branch of this hyperbola have certain configurations. To determine if a curve is a hyperbola, we can compare the similarity of the first and second derivative configurations of this curve with those of a predefined "south-opening" hyperbola with the related *normalized cross correlation* (NCC) values. As a typical hyperbola of response from buried utilities, $y^2/25-x^2/16=1$ is used as the predefined hyperbola in all the experiments in this work. By testing with different hyperbolae, we found that there is no significant difference if other hyperbolae are used as the predefined template hyperbola because NCC is invariant to scaling and the shape of hyperbolae do not change significantly for small-sized objects.

When x is discretized in a certain range, the related first and second derivatives of the predefined hyperbola make up two vectors, which are used as templates to identify the hyperbolic signatures from the outputs of the C3 algorithm. To use the templates, for each output cluster from the C3 algorithm, the central string is computed as shown in Fig. 6(b). NCC values of the first and second derivative values along each central string against the templates are computed after aligning the peaks of the central string and the predefined hyperbola curve. The NCC value of two vectors v_1 and v_2 is defined as follows:

$$\text{ncc}=\frac{|v_1\cdot v_2|}{|v_1|*|v_2|}. \tag{5}$$

When two "south-opening" hyperbolae are aligned with respect to the x coordinates of their centers, the NCC values of their first and second derivative curves are high (close to 1).

The NCC values of the first and second derivatives are used in the following neural network classification step to identify the hyperbolic signatures.

B. Neural Network Classification

A group of positive and negative samples is manually selected from the outputs of the C3 algorithm, and the two NCC values of each sample are computed, which are used to train a neural network classifier. This stage provides the subsequent stages with a continuous measure of confidence as to whether a particular output of the C3 algorithm is a hyperbolic signature or not. First, a three-layer feedforward perceptron neural network (as in Fig. 9) was trained with the backpropagation learning algorithm [35], and the corresponding vectors were recorded. The trained neural network can be applied to classify the outputs of C3 algorithms new to the neural network.

In practice, a smoothed version of the central string is used when comparing with the templates. Judged by the experimental results in Section V, the proposed neural network classification algorithm works very well for most hyperbolic signatures.

IV. Orthogonal-Distance Hyperbola Fitting

In this section, we present a robust orthogonal-distance fitting algorithm for hyperbola fitting [26] and introduce a method to initialize a hyperbola directly from given points.

A. Hyperbola Fitting Algorithm

Given a set of points $(x_i, y_i)_{i=1}^m$, the orthogonal distance d_i of a point $P_i = (x_i, y_i)$ to a hyperbola can be expressed by

$$d_i^2 = \min_{\phi_i} \left[(x_i - x(\phi_i))^2 + (y_i - y(\phi_i))^2 \right] \tag{6}$$

where $(x(\phi_i), y(\phi_i))$ is the corresponding closest point of P_i on the hyperbola.

The task is to determine a, b, x_0, and y_0 for this hyperbola by solving

$$\text{argmin}_{a,b,x_0,y_0} \sum_{i=1}^{m} d_i^2. \tag{7}$$

It is not a trivial task to find the closest point of P_i on a hyperbola when P_i itself is not on this hyperbola as explained below. Suppose $P(x, y)$ is the closest point of P_i on the hyperbola expressed by (2). Since the connecting line of P and P_i is perpendicular to the tangent line of the hyperbola at P, the coordinates of P satisfy the following equation:

$$\frac{dy}{dx} \cdot \frac{y_i - y}{x_i - x} = \frac{a^2(x - x_0)}{b^2(y - y_0)} \cdot \frac{y_i - y}{x_i - x} = -1. \tag{8}$$

The coordinates of P can be obtained by solving the system of (2) and (8) with a generalized Newton method. The average time needed for finding the closest point of a given point on a hyperbola is about 0.0015 s using a computer with an Intel 3.6-GHz processor coded in MATLAB.

After finding the closest point P on the hyperbola for each given point P_i, the coefficients of the hyperbola satisfying (7) can be obtained by using Gauss–Newton iteration, i.e.,

$$\mathbf{J} \cdot \Delta\mathbf{c} = \Delta\mathbf{P} \tag{9}$$

$$\mathbf{c}_{k+1} = \mathbf{c}_k + \lambda \Delta\mathbf{c} \tag{10}$$

where $\mathbf{c} = [a, b, x_0, y_0]^t$ are the parameters of the current hyperbola, $\Delta\mathbf{P} = |\mathbf{P} - \mathbf{P_i}|$ with $\mathbf{P_i} = [x_i, y_i]^t$, a given point, and $\mathbf{P} = [x, y]^t$, the closest corresponding point of $\mathbf{P_i}$ on the current hyperbola. $\mathbf{J} = (\partial F/\partial c)|_{\mathbf{c}_k}$ is the Jacobian matrix with F as the corresponding expression of the current hyperbola, and λ is the step size parameter.

B. Direct Hyperbola Initialization

In previous work on orthogonal-distance fitting, some authors suggest taking the initial parameter values from the corresponding algebraic distance fitting [26], [29]. In this paper, because of the robustness of the fitting algorithm and the fact that we only deal with the south-opening branch of a hyperbola from GPR data, we propose a simple and fully automatic procedure to directly compute the parameters of the initial hyperbola, which works very well for converging to the global minimum of (7) in our experiments.

To determine a south-opening branch of a hyperbola, if its apex is given, only two other points, which satisfy certain constraints, are needed.

Given (x_v, y_v) as the coordinates of the apex of a south-opening branch of a hyperbola and (x_l, y_l) as a point on the left-hand side of line $x = x_v$ and (x_r, y_r) as a point on the right-hand side of line $x = x_v$, what constraints must be satisfied to determine a hyperbola? Clearly, the following two constraints should be satisfied first: $y_v > y_l$ and $y_v > y_r$. Second, (x_l, y_l) and (x_r, y_r) cannot be symmetric to the line of $x = x_v$. The reason will be given later in this section. Third, when x_v, y_v, x_l, y_l, and x_r are fixed values, the value of y_r must satisfy the following equations:

$$y_r < y_v + \frac{(x_v - x_r) \cdot (y_v - y_l)}{x_v - x_l} \tag{11}$$

$$y_r > \frac{s_l \cdot y_v - s_r \cdot (y_v - y_l)}{s_l} \tag{12}$$

where $s_r = (x_r - x_v)^2$, and $s_l = (x_l - x_v)^2$.

For a given set of points $(x_i, y_i)_{i=1}^m$ for fitting a south-opening hyperbola, to initialize a hyperbola, we first compute three points from the given points. First, the point with the largest y-coordinate is found, and a centroid is computed among the given points within a neighborhood of this point. This centroid is used as the apex of the initial hyperbola. Denote its coordinates as (x_v, y_v), then $x_0 = x_v$ with x_0 as the x-coordinate of the center of the initial hyperbola. Then, pick a region to the left of (x_v, y_v), which includes some given points. Denote the coordinates of the centroid of the given points within this region as (x_l, y_l), which are also used to compute the initial hyperbola. The same procedure is applied to the right side of (x_v, y_v) to obtain a point (x_r, y_r). To avoid (x_l, y_l) and (x_r, y_r) being symmetric to line $x = x_v$, the regions picked on both sides of (x_v, y_v) should have different distances to line $x = x_v$. If the value of y_r satisfies (11) and (12), its value is used to initialize the hyperbola; otherwise, its value is replaced by the average of the right-hand sides of (11) and (12). Then, the other three parameters in (2) can be computed as follows:

$$y_0 = \frac{s_l \cdot y_r^2 - s_r \cdot y_l^2 + (s_r - s_l) \cdot y_v^2}{2(y_r \cdot s_l - y_l \cdot s_r + y_v \cdot (s_r - s_l)} \tag{13}$$

$$a^2 = (y_0 - y_v)^2 \tag{14}$$

$$b^2 = \frac{s_r \cdot a^2}{(y_r - y_0)^2 - a^2} \tag{15}$$

where s_r and s_l are the same as in (11) and (12), and y_0 is the y-coordinate of the center of the initial hyperbola.

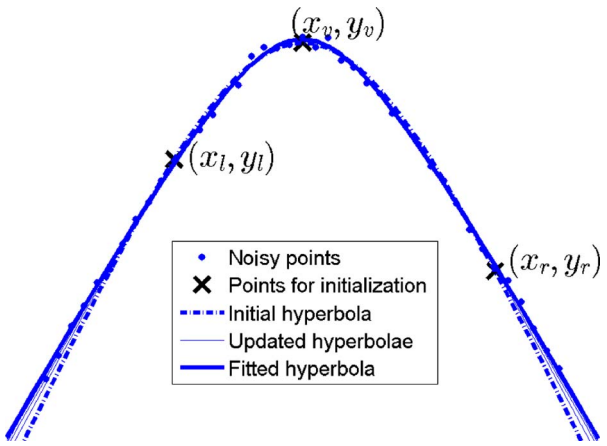


Fig. 10. Example of hyperbola fitting with a synthetic data set.

If (x_l, y_l) and (x_r, y_r) are symmetric to line $x = x_v$, then $s_r = s_l$, and $y_r = y_l$. In this situation, the denominator on the right-hand side of (13) is zero. Hence, points (x_l, y_l) and (x_r, y_r) should not be symmetric to line $x = x_v$.

An example of orthogonal-distance hyperbola fitting is presented in Fig. 10 where the initial hyperbola is computed with the proposed method. Although only three points are used to compute the initial hyperbola, it is reasonably close to the given points. After sufficient steps, the fitting procedure converges. In our experiments, most fittings converge within 100 iterations.

V. Experiments

In this section, experimental results on synthetic and real data are displayed and analyzed. The computational cost is also analyzed in this section.

A. Synthetic Data

First, we applied the proposed algorithm on synthetic data sets. The synthetic data sets are generated to simulate the different scenarios of hyperbola configuration in GPR images, such as hyperbolae with different shapes and sizes, intersecting hyperbolae with crossing legs, noisy strips and points, etc. In the first experiment (see Fig. 11), there are two hyperbola-shaped regions and three linear segment regions [see Fig. 11(a)]. There is no intersection between the two hyperbolic signatures, but one of the linear segments is connected to one of the hyperbolic signatures. There are five clusters in total given by the C3 algorithm, and four of them are displayed in Fig. 11(b) and (c). From the output clusters, it can be seen that the hyperbolic signature, which is connected to a linear segment region, is separated from it [see Fig. 11(c)].

In the second experiment (see Fig. 6), apart from the connections of a hyperbolic signature with the linear segment regions, there is an intersection between the hyperbolic signatures in the input image. The experimental result demonstrates that the hyperbolic signatures can be clearly separated from each other by the C3 algorithm [see Fig. 6(c)]. In Fig. 6(b) and (c), the central string of the corresponding clusters is also displayed. In our experiments, a smoothed version of each central string is used in the neural network classification algorithm for identifying hyperbolic signatures.

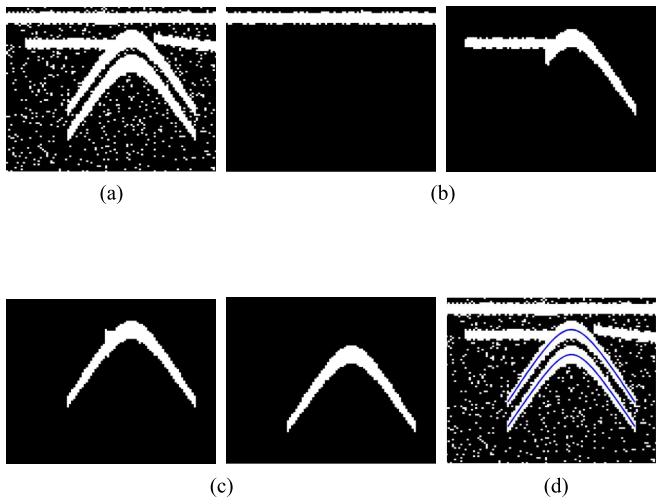


Fig. 11. Illustration of the application of the proposed system on a synthetic data set. (a) Input image. (b) Linear or hybrid clusters. (c) Hyperbola-shaped clusters. (d) Output image.

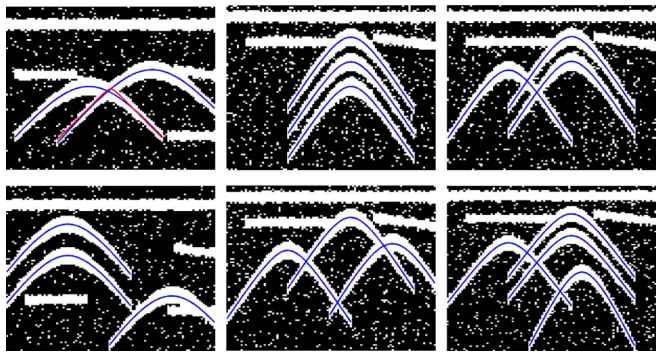


Fig. 12. Some experimental results on synthetic data (best viewed in color).

TABLE I
Experimental Results on Synthetic Data

Ground truth	True positive	False positive
52	52	3

More experimental results on synthetic data sets are displayed in Fig. 12. It can be seen that all the hyperbolic signatures are detected. In the synthetic data, there are many intersections between different hyperbola branches. For each such intersection, a cluster is obtained through the C3 algorithm such as that displayed in the right image of Fig. 6(b). Most of them are correctly classified by the neural network classification algorithm as nonhyperbolic signatures, and only few of them are regarded as hyperbolic signatures such as the red curve in the first image of Fig. 12. Precise statistics are given in Table I. For the correctly classified hyperbolic signatures, the proposed hyperbola fitting algorithm converges to the global minimum in all cases.

B. Real Data

We also applied our algorithms to real data sets. For a real data set, a thresholding step needs to be applied to separate

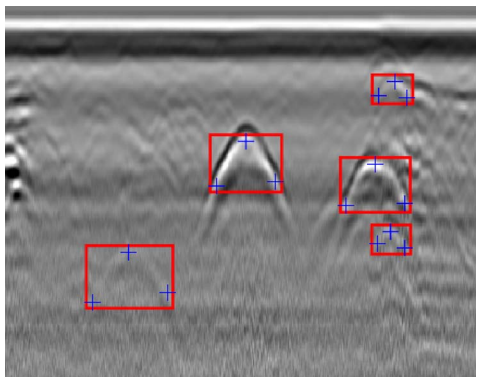


Fig. 13. Example of ground truth in GPR images (best viewed in color).

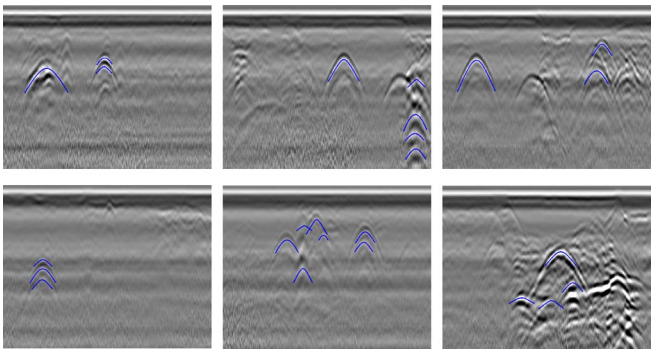


Fig. 14. Some experimental results on real data (best viewed in color).

the regions of high response from the background (see Fig. 4). After this step, the remaining procedures are the same as those of the synthetic data.

In the real data set, 100 GPR images were collected from an externally provided data set. The images contain hyperbolae at different depths; some of them are clear and well shaped, whereas some are weakly contrasted and asymmetric with numerous interactions between each other. From these images, 464 hyperbolae were manually annotated. They are used as the ground truth for training and testing. With this group of real data set, tenfold cross evaluation was performed. More details on the experimental results are given in the following sections.

To facilitate the evaluation of the experimental results with the ground truth, we use a simple way to represent hyperbolae in the ground truth. For each hyperbola in the ground truth, three points are marked manually: the apex, one point on the left-hand side of the apex, and another point on the right-hand side of the apex (see Fig. 13). All the coordinates of the marked points are recorded in a text file with respect to different images. For a fitted hyperbola in a certain test image, if a group of three marked points for ground truth is found with an average distance to that hyperbola of less than 10 pixels, this hyperbola is regarded as a true positive; otherwise, it is taken as a false positive.

Some experimental results on real data sets with the proposed method are displayed in Figs. 2, 14, and 16. There are 57 clusters given by the C3 algorithm in the experiment displayed in Fig. 2 and 45 clusters in the experiment displayed in Fig. 16.

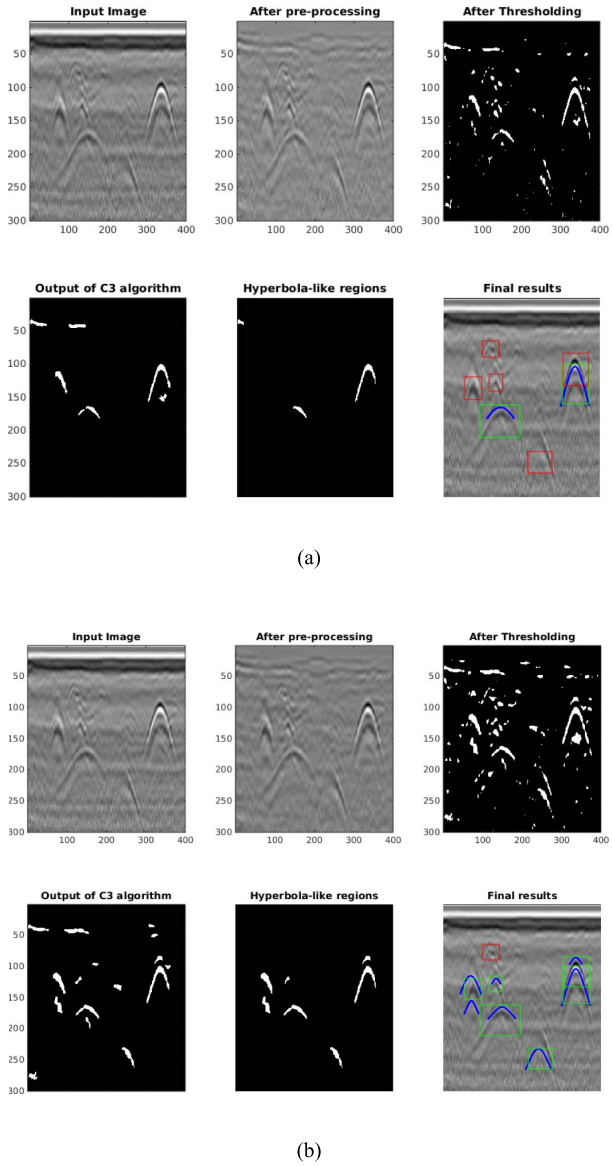


Fig. 15. Comparison of different thresholding methods on one real GPR image. (Blue curves) Fitted hyperbolae. (Green rectangles) Correctly found hyperbolae. (Red rectangles) Hyperbolae missed by the detection algorithm (best viewed in color). (a) Results with unimodal thresholding. (b) Results with the proposed thresholding.

Compared with the synthetic data, the real data are much more noisy. Hence, there are more output clusters in the real data experiments. The preprocessing step captured regions including most of the expected hyperbolae, and the neural network classification algorithm works effectively in picking most of the expected hyperbolic signatures for hyperbola fitting. For a comparison, experiments are repeated by replacing the proposed thresholding method with the unimodal thresholding method introduced in [34]. It is shown in Fig. 15 that the thresholding method proposed in this paper can keep more hyperbola regions than unimodal thresholding. Detailed statistics are given in Table II. It can be seen that among the ten random trials, the average detection rate and precision rate of the proposed method are higher than those of the unimodal thresholding method (see Fig. 15).

TABLE II
COMPARISON OF THE AVERAGE DETECTION RATES AND FITTING RATES AMONG DIFFERENT METHODS

Method	Recall	Precision	F-Measure
Detection rates of [9]	0.724	0.347	0.474
Fitting rates of [9]	0.418	0.091	0.149
Fitting rates of [34] + C3	0.638	0.654	0.643
Fitting rate of our method	0.704	0.708	0.702

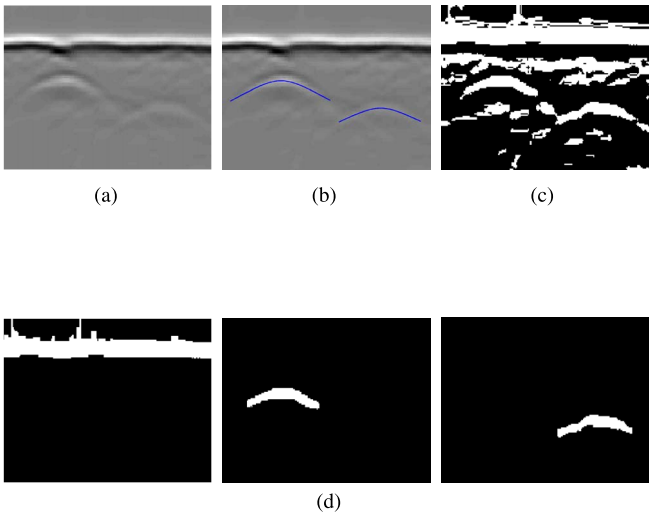


Fig. 16. Illustration of the application of the proposed technique on a real GPR image. (a) Input image. (b) Output image. (c) Regions of interest. (d) Some clusters obtained from the C3 algorithm (not all clusters from the C3 algorithm are shown here; best viewed in color).

C. Comparison of Hyperbola Detection and Fitting Rates With Another Method

We also compared the proposed method with that introduced in [9]. In [9], a Viola–Jones-based detector is used to detect the candidate hyperbola regions at first, and then, a generalized Hough transform is used to extract hyperbola parameters by fitting the hyperbolic edges of each candidate region. As it did not provide any details on the Hough-transform-based hyperbola fitting results, we compare this method with our proposed method using two metrics: the detection rate and the fitting rate. As previously mentioned, if a group of three marked hyperbola points for ground truth is found with an average distance to a fitted hyperbola of less than 10 pixels, the fitted hyperbola is regarded as a true positive; otherwise, it is taken as a false positive. Since, in the proposed method, if a cluster is identified as a hyperbolic signature, a hyperbola is always fitted to that region, the detection rate and the fitting rate of the proposed method are the same. However, an obvious difference can be found in the detection rate and fitting rate with the methods proposed in [9].

1) Detection Rate: When we mark the three points for each hyperbola in the ground truth, a rectangular bounding box with its sides parallel to the axes is generated with the marked points. An enlarged rectangle window with 5-pixel offset from each side of the bounding box is also recorded, as shown in Fig. 13 for later use. For the method in [9], if a detected region has more than 60% overlap with any recorded rectangle window in the ground truth, it is considered as a correct detection. For training purposes, the rectangle windows related to training images are also used to crop the hyperbola regions and saved as positive

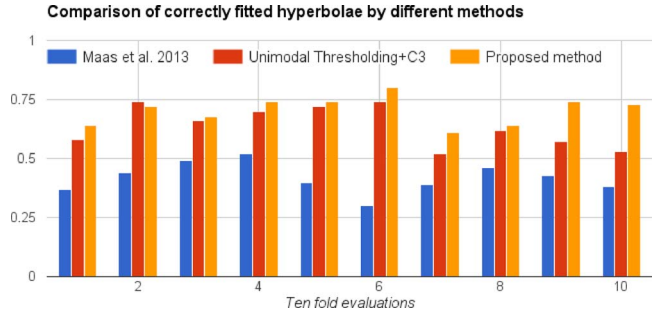


Fig. 17. Comparison of the fitting rates of different methods on ten cross evaluations.

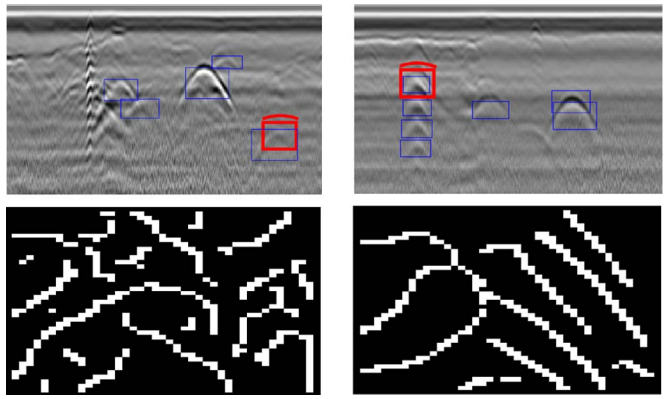


Fig. 18. Generalized Hough transform fails to fit correct hyperbolae in some detected regions. First row: input GPR images with the ground truth marked by blue rectangles. The red curve is the fitted hyperbola to the region in the red window based on the edge points as shown in the second row. It can be seen that the fitted hyperbolae are outside the detected region and are regarded as false fitting (best viewed in color).

samples. From the background regions, 3000 negative samples are also randomly generated. When using the Haartraining package of OpenCV to train the classifier, a basic resolution of 24 $\times$ 24 pixels of each region is used in the training procedure. Then, the obtained classifier is used to detect candidate regions in the testing data set. Detailed statistics of the average detection rate of the method used in [9] are given in the first row of Table II. It can be seen that the detection rate is 0.72, but the average precision rate is only 0.35 (see Table II).

2) Fitting Rate of A Generalized Hough Transform Based Method: As presented in [9], the candidate regions are smoothed with a Gaussian filter to reduce noise and artifacts and then converted with a Canny edge detector into a binary image. After that, a generalized Hough transform is used to fit hyperbolae based on the edge points. For each candidate region, only the best hyperbola given by the generalized Hough transform is fitted. Each fitted hyperbola is then compared with the ground truth with the same criterion as previously described. It can be seen in some cases that even when a correct region is detected in the detection step, the generalized Hough transform fails to fit the correct hyperbola, as shown in Fig. 18. The recall rates of the correctly fitted hyperbola from different methods are shown in the second column of Table II and Fig. 17. In Fig. 17, the two images on the first row are only used to demonstrate the original GPR image overlapped with the detected candidate regions from the Viola–Jones-based detector, and the figures on the second row are the enlarged windows of the red rectangles in the images on the first row. The top horizontal

TABLE III
COMPUTATION TIME OF HYPERBOLA FITTING USING A COMPUTER WITH AN INTEL 3.6-GHz PROCESSOR

Method	**Fitting time per hyperbola (s)**	
	Discretization values of d_s	Time
Generalized Hough transform based fitting method	$ds = (1, 1, 1, 1)$	2.97
	$ds = (1, 1, 0.5, 0.5)$	11.4
	$ds = (0.5, 0.5, 0.5, 0.5)$	47.3
	$ds = (0.2, 0.2, 0.2, 0.2)$	895.3
Proposed hyperbola fitting method		**0.73**

pattern of the GPR image has no influence on the generalized Hough transform results since each candidate region was then cropped and processed separately for hyperbola fitting.

D. Computational Time

The size of the synthetic input images is 100×100 pixels. The average computational time of the experiments on one sample image using a computer with an Intel 3.6-GHz processor is approximately 0.43 s. The computational time on real images depends on how many hyperbolae are detected. In our experiments, the sizes of the real input images are 300×400 pixels, and the computational time of a real image is, on average, $0.48 + 0.73 \times n$ s with n being the number of candidate hyperbolae for fitting; on average, six hyperbolae were detected and fitted in each test image. This speed is fast enough for real-time on-site applications.

As previously mentioned, the computation time of the generalized Hough-transform-based hyperbola fitting method highly depends on the discretization of the parameters. Table III shows the computation time of the generalized Hough transform method when only changing the discretization of the parameters (a, b, x_0, y_0) of (1). In this table, $ds = (d_a, d_b, d_{x_0}, d_{y_0})$ denotes the discretization steps of the parameters. It can be seen that when $ds = (0.2, 0.2, 0.2, 0.2)$, the average computation time is about 15 min, which is not comparable with the proposed method.

VI. CONCLUSION AND FUTURE WORK

In this paper, a novel technique for the automatic interpretation of GPR images has been introduced. The proposed system[1] allows for the detection of the presence of underground buried objects and can obtain hyperbola parameters by fitting a hyperbola to each hyperbolic signature in a completely automatic manner. The C3 algorithm is based on the connecting elements from adjacent columns of the image, which is different from conventional distance/density-based clustering techniques. It cannot only cluster the separated hyperbolic signatures but also segment intersected or connected hyperbolic signatures into separated ones. The neural network classification algorithm for identifying hyperbolic signatures needs only two features and can be easily trained with a small set of training data. The orthogonal-distance hyperbola fitting algorithm is robust and efficient for fitting "south-opening" hyperbolae. The hyperbola parameters obtained through the orthogonal-distance fitting algorithm can be used in further applications such as estimating the size of the objects [12]. Despite the intrinsic complexity of GPR images, the experimental results show that the proposed method exhibits very good performance compared with a state-of-the-art method, in terms of robustness to noise, efficiency, and accuracy, and is fast enough for real-time on-site applications. The proposed thresholding method works very well compared with other classic thresholding methods; however, we believe that a "multilevel thresholding" method, which we are currently studying, may improve the current method even further by adaptively segmenting the weak reflections such as those from small plastic pipes.

[1]The code will be placed in an open source repository.

REFERENCES

[1] D. Daniels, *Ground Penetrating Radar*. London, U. K.: The Inst. Elect. Eng., 2004.

[2] C. Bruschini, B. Gros, F. Guerne, P. Y. Pièce, and O. Carmona, "Ground penetrating radar and imaging metal detector for antipersonnel mine detection," *J. Appl. Geophys.*, vol. 40, no. 1–3, pp. 59–71, 1998.

[3] G. Ciochetto, S. Delbo, P. Gamba, and D. Roccato, "Fuzzy shell clustering and pipe detection in ground penetrating radar data," in *Proc. IGARSS*, 1999, vol. 5, pp. 2575–2577.

[4] E. Costamagna, P. Gamba, and E. Lossani, "A neural network approach to the interpretation of ground penetrating radar," in *Proc. IGARSS*, 1998, vol. 1, pp. 412–414.

[5] P. Gamba and S. Lossani, "Neural detection of pipe signatures in ground penetrating radar images," *IEEE Trans. Geosci. Remote Sens.*, vol. 38, no. 2, pp. 790–797, Mar. 2000.

[6] R. Janning, T. Horvath, A. Busche, and L. Schmidt-Thieme, "GamRec: A clustering method using geometrical background knowledge for GPR data preprocessing," *Artif. Intell. Appl. Innov., IFIP Adv. Inf. Commun. Technol.*, vol. 381, pp. 347–356, 2012.

[7] R. Janning, A. Busche, T. Horvath, and L. Schmidt-Thieme, "Buried pipe localization using an iterative geometric clustering on GPR data," *Artif. Intel. Rev.*, vol. 42, pp. 403–425, 2013.

[8] R. Janning, T. Horvath, A. Busche, and L. Schmidt-Thieme, "Pipe localization by apex detection," in *Proc. IET Int. Conf. Radar Syst.*, 2012, pp. 1–6.

[9] C. Maas and J. Schmalzl, "Using Pattern Recognition to Automatically Localize Reflection Hyperbolas in Data From Ground Penetrating Radar," vol. 58, pp. 116–125, Aug. 2013.

[10] L. Mertens, R. Persico, L. Matera, and S. Lambot, "Automated detection of reflection hyperbolas in complex GPR images with no a priori knowledge on the medium," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 1, pp. 580–596, Jan. 2016.

[11] E. Pasolli, F. Melgani, and M. Donelli, "Automatic analysis of GPR Images: A pattern-recognition approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 7, pp. 2206–2217, Jul. 2009.

[12] S. Shihab and W. Al-Nuaimy, "Radius estimation for cylindrical objects detected by ground penetrating radar," *Sens. Imaging, Int. J.*, vol. 6, no. 2, pp. 151–166, 2005.

[13] J. Illingworth and J. Kittler, "A survey of the Hough transform," *Comput. Vis., Graph., Image Process.*, vol. 44, no. 1, pp. 87–116, 1988.

[14] L. Capineri, P. Grande, and J. A. G. Temple, "Advanced image-processing technique for real-time interpretation of ground-penetrating radar images," *Int. J. Imaging Syst. Technol.*, vol. 9, no. 1, pp. 51–59, 1998.

[15] P. Falorni, L. Capineri, L. Masotti, and G. Penilli, "3-D radar imaging of buried utilities by features estimation of hyperbolic diffraction patterns in radar scans," in *Proc. 10th Int. Conf. GPR*, 2004, vol. 1, pp. 403–406.

[16] C. G. Windsor, L. Capineri, and P. Falorni, "A data pair-labeled generalized Hough transform for radar location of buried objects," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 1, pp. 124–127, Jan. 2014.

[17] M. Fritze, "Detection of buried landmines using ground penetrating radar," in *Proc. SPIE*, 1995, vol. 2496, pp. 100–108.

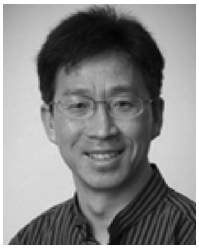
[18] W. Alnuaimy, Y. Huang, M. Nakhkash, M. Fang, V. Nguyen, and A. Erisken, "Automatic detection of buried utilities and solid objects with GPR using neural networks and pattern recognition," *J. Appl. Geogr.*, vol. 43, pp. 157–165, 2000.

[19] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, 2004.

[20] A. Giannopoulos, "Modelling ground penetrating radar by gprMax," *Construct. Build. Mater.*, vol. 19, no. 10, pp. 755–762, 2005.

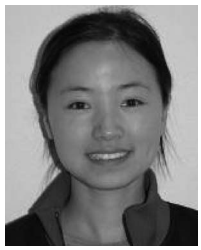
[21] H. Chen and A. G. Cohn, "Probabilistic robust hyperbola mixture model for interpreting ground penetrating radar data," in *Proc. IEEE World Congr. Comput. Intell.*, 2010, pp. 3367–3374.

[22] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 881–892, Jul. 2002.
[23] R. Ng and J. Han, "Efficient and effective clustering methods for spatial data mining," *Proc. 20th Int. Conf. Very Large Data Bases*, 1994, pp. 144–155.
[24] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. KDD*, 1996, pp. 226–231.
[25] L. Ertöz, M. Steinbach, and V. Kumar, "Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data," in *Proc. 2nd SIAM Int. Conf. Data Mining*, 2003, pp. 1–12.
[26] S. Ahn, W. Rauh, and H.-J. Warnecke, "Least-squares orthogonal distance fitting of circle, sphere, ellipse, hyperbola and parabola," *Pattern Recognit.*, vol. 34, no. 12, pp. 2283–2303, Dec. 2001.
[27] F. L. Booktein, "Fitting conic sections to scattered data," *Comput. Graph. Image Process.*, vol. 9, no. 1, pp. 56–71, 1979.
[28] A. Fitzgibbon, M. Pilu, and R. B. Fisher, "Direct least square fitting of ellipses," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 476–480, May 1999.
[29] W. Gander, G. Golub, and R. Strebel, "Least-squares fitting of circles and ellipses," *BIT*, vol. 34, no. 4, pp. 558–578, Dec. 1994.
[30] M. Pilu, A. Fitzgibbon, and R. B. Fisher, "Ellipse-specific direct least-square fitting," in *Proc. Int. Conf. Image Process.*, 1996, vol. 3, pp. 1–4.
[31] H. Chen and A. G. Cohn, "Probabilistic conic mixture model and its applications to mining spatial ground penetrating radar data," in *Proc. Workshops SDM*, 2010, pp. 1–9.
[32] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man Cybern.*, vol. 9, no. 1, pp. 62–66, Jan. 1979.
[33] J. Kapur, P. Sahoo, and A. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram," *Comput. Vis., Graph., Image Process.*, vol. 29, no. 3, pp. 273–285, 1985.
[34] P. L. Rosin, "Unimodal thresholding," *Pattern Recognit.*, vol. 34, no. 11, pp. 2083–2096, Nov. 2001.
[35] J. M. Zurada, *Introduction to Artificial Neural Systems*. New York, NY, USA: West, 1992.



Qingxu Dou received the M.Sc. degree in applied mathematics and the Ph.D. degree in electric and electronic engineering from Heriot-Watt University, Edinburgh, U.K., in 2006 and 2011, respectively.

He is currently a Research Fellow with the School of Computing, University of Leeds, Leeds, U.K. His research areas of interest are computer vision, sensor data interpretation, data fusion, buried utility mapping, and mathematical modeling.



Lijun Wei received the B.Sc. degree in geography science from Wuhan University, Wuhan, China, in 2008, and the Ph.D. degree in computer science from the University of Technology of Montbéliard, Belfort, France in 2013.

She is a Research Fellow in Computer Science with the University of Leeds, Leeds, U.K. Her main areas of interest are image processing and multisensor fusion particularly on mapping and intelligent vehicle localization/navigation.

Derek R. Magee received the B.Sc. degree in engineering (first class) from Durham University, Durham, U.K., in 1995 and the Ph.D. degree from the University of Leeds, Leeds, U.K., in 2001.

He is a Lecturer in computer science with the University of Leeds. He is also a Chief Technical Officer with the Medical Image Analysis company HeteroGenius Ltd. His main areas of interest are image analysis and machine learning particularly on medical image and remote sensing data.

Anthony G. Cohn received the B.Sc. and Ph.D. degrees in computer science from the University of Essex, Essex U.K.

He is currently a Full Professor with the School of Computing, University of Leeds, Leeds, U.K. His research interests include artificial intelligence, knowledge representation and reasoning, cognitive vision, robotics, sensor fusion, and decision support systems. For more than a decade part of his research has focused on street works and utilities. The VAULT system, which provides 24/7 real-time integrated utility data across Scotland arising from his Mapping the Underworld and VISTA projects, won a 2012 Institution of Engineering and Technology Innovation Award and the 2012 National Joint Utilities Group Award for "Avoiding Damage."

Dr. Cohn is a Fellow of the Royal Academy of Engineering, the Association for Advancement of Artificial Intelligence, and the European Association for Artificial Intelligence. He has received Distinguished Service awards from the International Joint Conferences on Artificial Intelligence and the Association for the Advancement of Artificial Intelligence.