Automated Defect Detection and Visualization for the Robotic Airport Runway Inspection

ZHONGCHENG GUI1 AND HAIFENG LI2
1Chengdu Guimu Robot Company, Ltd., Chengdu 610101, China
2College of Computer Science and Technology, Civil Aviation University of China, Tianjin 300300, China
Corresponding author: Haifeng Li (hfli@cauc.edu.cn)

This work was supported by the National Key Research and Development Project of China under Grant 2019YFB1310601 and Grant 2019YFB1310605.

ABSTRACT Detection of both surface and subsurface defects is a vital task for maintaining the structural health and reliability of airport runways. We report the automated data collection and analysis for airport runways based on our novel robotic system, which employs a camera and a GPR (Ground Penetrating Radar) to inspect the surface and subsurface conditions, respectively. To perform the automated data analysis, we propose a novel crack detection algorithm based on the images, and a subsurface defect detection method with GPR data. Additionally, to create a composite global view of a large airport runway span, a camera/GPR data sequence from the robot is aligned accurately to create a continuous mosaic for visualization. We combine these algorithms into a software to perform automated on-site analysis. We have put our robot and software into engineering practice over 20 airports in China, achieving the performance of 70% and 67% F1-measure for crack detection and subsurface defect detection, respectively. More importantly, the results of our algorithms can satisfy the requirement of applications.

INDEX TERMS Airport runway inspection, crack detection, subsurface defect detection, image stitching.

I. INTRODUCTION

Structural defect inspection is essential task to monitor the health and reliability of the airport runways. Early detection of defects, such as surface cracks and subsurface voids, for airport runway is an important maintenance task. By the end of 2019, there are 238 civil transport airports in China, with nearly 300 runways. Structural defects exist in many airport runways, leading to costly maintenance even safety risks if serious defects cannot be detected in their early stages.

Traditional manual defect detection methods are very labor-intensive, time-consuming, with low-accuracy, and error prone. Even slight defects may be early warning signs of significant failure for airport runways, and need to be detected accurately and timely. Thus, it is necessary to develop an automated defect detection method for airport runway inspection.

We develop an airport runway inspection robot (ARIR), with a camera and a GPR (Ground Penetrating Radar) fixed to perform the condition sensing task. This paper focuses on addressing the problem of analyzing the collected data accurately and automatically. Specifically, we propose a robust image-based crack detection algorithm, and a deep learning based subsurface defect detection method from GPR data. In addition, to create a composite global view of a large airport runway span, a camera/GPR data stitching algorithm is presented to create a continuous mosaic for visualization.

The rest of paper is organized as follows: the related work is summarized in Section II before we introduce our sensing suite configuration and data collection scheme in Section III. The proposed algorithms are illustrated in details in Section IV. The performance of our algorithms is validated in experiments in Section V, and the paper is concluded in Section VI.

II. RELATED WORK

In recent years, research in structure inspection using robotic systems has attracted much attention and resulted in several prototypes, such as the bridge deck inspection robot [1], [2], and the robot for tunnel structure health inspection [3]. However, the scenario of airport runway inspection is quite different and more challenging. We develop a robotic system for airport runway inspection. In order to implement the automated data analysis for airport runway inspection, there
are three challenging problems, including large scale image stitching, crack detection from images, and defect detection from GPR data, that need to be addressed.

General image stitching techniques have been well studied [4]–[6] and even put into commercial use. The homography model is adopted in these methods and point features in images are used to estimate the homography model [7], [8]. However, the image stitching problem in airport runway inspection is significantly different in two points: On the one hand, the surveyed regions is very large, involving huge amount of images to be stitched. On the other hand, images of airport runway are usually lack of features, leading to occasional but inevitable failures of feature-based homography estimation.

Existing image-based crack detection methods can be mainly classified into two categories: the standard image processing methods, and machine learning based techniques. Standard image processing methods, including intensity thresholding [9], edge detection [10], and morphological filtering [11], [12], have been widely studied for automated crack detection. However, the performance of these methods is usually dependent upon the parameter choices [13] which are very difficult to accomplish for field images with significant visual clutters, leading to unreliable detection results in airport runway inspection applications. Machine learning based crack detection methods build on techniques such as support vector machines [14], random forest [15], random structured forest [16], and neural networks [17]. The machine learning based methods obtain more robust performance compared with image processing techniques. However, a supervised training stage is needed which requires large amount of labeled data, a difficult requirement for applications in airport runway since the sample images with cracks are rare. Our proposed crack detection algorithm combines the advantages of image processing methods in few samples and machine learning based techniques in robustness, achieving a satisfying performance on airport runway data.

Recognizing the subsurface objects automatically from GPR data is nontrivial, because a GPR cannot provide 3D positions but a reflection image full of significant signal clutters. Thus, object detection from GPR in an automated manner is still a challenging problem. Standard signal processing methods, such as template matching [18], S-transform [19], and wavelet transform [20], have been used in GPR data analysis. However, these methods are sensitive to noise, so cannot be employed directly in subsurface defect detection. Machine learning based methods, especially the CNN-based deep learning techniques [21], [22] obtain much attention in GPR data analysis [23], [24]. Although these CNN-based techniques have achieved a certain of positive results, they still cannot satisfy the requirement of field applications. One of the main reasons is that only 2D B-scan images are employed, which ignores the natural 3D property of subsurface defects.

Difference with all of the above mentioned works, our paper focuses on the airport runway data analysis which is collected by our ARIR system integrated with camera and GPR. The developed data analysis algorithms allows the robot to detect both surface and subsurface defects robustly, and build the large scale surveyed airport runway image to perform high-efficient assessments.

### III. SENSOR CONFIGURATION AND DATA COLLECTION

This section firstly presents the sensor configuration of our ARIR system. The sensor setup of the ARIR system is shown in Fig. 1, which consists of one Raptor GPR with 900MHz antennas, and one DALS Nano M1920 camera with a resolution of 1920 × 1200.

To conduct defect inspection, the robot navigates within a predefined surveyed region on the airport runway to collect images and GPR data, as shown in Fig. 2. The robot first moves from the start to point A, then follows a linear path along each scan. The sensing suite can cover 1 meter wide in each scan. Once the robot finish the current scan, it moves to the next scan until the entire surveyed region is completely covered. At the end of each scan, the robot first moves to the turn point along an omni path, and then moves to the other scan to ensure there is no any region missed out. When scanning, the robot simultaneously transfers the image and GPR data to the nearby data analysis center in a van using 4G/5G connection. Then the collected data will be automatically analyzed off-line, which is presented in the following section.

### IV. AUTOMATED DATA ANALYSIS AND VISUALIZATION METHODS

#### A. LARGE SCALE IMAGE STITCHING FOR VISUALIZATION

There are two most challenging problem in our large scale image stitching. One is the large amount of images, easily causing the error accumulation; The other one is the lack of matched features between some adjacent images. To handle these problems, a two-stage image stitching algorithm is proposed. In this first stage, we align all images according to their residing GPS positions. Then in the second stage, we perform the adjustment under a global optimization using the available matched point features.
1) STAGE 1: POSITION BASED INITIAL STITCHING
Our robot has synchronized the RTK GPS and camera. Thus we can record global position for each image. Generally, the localization error of GPS while working in the open airport runway is about 5cm. Since we fix the camera downward to the ground, the image plane is approximately parallel to the horizontal airport runway plane. We build a world coordinate system \( \{W\} \) with \( X - Y \) plane to be horizontal. Then we align each image in the \( X - Y \) plane of \( \{W\} \) to form the initial stitching. Denote the resolution for each image as \( f_x \times f_y \) pixels, and each image covers a region of \( w_x \times w_y \) meters in the \( X - Y \) plane of \( \{W\} \). Due to the GPS localization error, the initial stitching result need to be optimized in the following stage.

2) STAGE 2: FEATURE BASED REFINEMENT
In this stage, we first extract the matched point features between adjacent overlapped images. Then we define a cost function to perform global refinement.

Denote \( I_a, a = 1, \ldots, N_I \) as the \( a \)-th image, with \( N_I \) being the total number of images. We define \( I_a \) and \( I_b \) to be neighbors if their overlapping ratio is bigger than 0.2 in the original stitching result. Then we extract all matched points between \( I_a \) and \( I_b \) using SURF [25] method. The homography model is adopted to filter out the mismatched feature points under RANSAC (Random Sample Consensus) [26] framework. We define \( \mathbf{x}_i^a \) and \( \mathbf{x}_i^b \) to be homogeneous coordinates of two matched SURF points between two neighbored images \( I_a \) and \( I_b \). A homography aims to map \( \mathbf{x}_i^b \) to \( \mathbf{x}_i^a \) following the relation

\[
\mathbf{x}_i^a = \mathbf{H}_{a,b} \mathbf{x}_i^b, \tag{1}
\]

where \( \mathbf{H}_{a,b} \) is the homography model between \( I_a \) and \( I_b \). Thus the geometric distance between \( \mathbf{x}_i^a \) to \( \mathbf{x}_i^b \) can be computed as

\[
d_i = \| \mathbf{x}_i^a - \mathbf{H}_{a,b} \mathbf{x}_i^b \|. \tag{2}
\]

With the distance defined in (2), we find the outliers as the mismatched points iteratively under RANSAC framework. With the mismatched points removed, we keep the correctly matched SURF points for the following refinement.

In theory, the matched points from different images should be at the same position in the stitched image. Due to the errors of feature points, for each image \( I_a \), we combine all matched points with its neighbored images to compute the displacement for refinement. The displacement for \( I_a \) is defined as

\[
d_a = \frac{1}{M_b S} \sum_{j=1}^{M_b} \sum_{i=1}^{S} \| \mathbf{x}_i^a - \mathbf{x}_i^{g(I_a,j)} \|, \quad \text{with} \quad \mathbf{x}_i^a \leftrightarrow \mathbf{x}_i^{g(I_a,j)} \tag{3}
\]

where \( M_b \) denotes the total number of \( I_b \)'s neighbors, \( g(I_a, j) \) is a function returning the image index number of the \( j \)-th neighbor of \( I_a \), \( S \) denotes the total number of matched feature points between \( I_a \) and \( I_b(g(I_a, j)) \), the symbol \( \leftrightarrow \) indicates one point corresponds to the other one.

Thus, the global refinement for image stitching can be performed by solving the following optimization problem,

\[
\arg \min_{\hat{p}_b, b = 1, \ldots, N_I} \sum_{a=1}^{N_I} d_a, \tag{4}
\]

where \( \hat{p}_b \), \( b = 1, \ldots, N_I \) is the estimated position of image \( I_b \).

We summarize our larege scale image stitching algorithm in Algorithm 1.

B. CRACK DETECTION FROM 2D IMAGES
In a gray scale 2D pavement image, cracks can be visually distinguished from the rest of the image, because of their lower intensity values and their shape of continuous long curvilinear shapes. Based on these two characteristics, an automatic crack detection algorithm is proposed.

1) INTENSITY BASED CANDIDATE CRACK PIXEL EXTRACTION
First, the intensity property is used to extract the candidate crack pixels from the background. The image is first smoothed by a mean smoothing filter. We perform variations on the standard mean smoothing filter by means of threshold averaging, wherein smoothing is applied subject to the condition that the center pixel value is changed only if the difference between its original value and the average value is greater than a preset threshold. This has the effect that noise is smoothed with a less dramatic loss in crack like details.
For each pixel \( x_i \), it will be kept as one candidate crack pixel if the following criteria is satisfied,

\[
g_{o}(x_i) \leq g_{i}(x_i) + T_g,
\]

where \( g_{o}(x_i) \) is the intensity of the original pixel, \( g_{i}(x_i) \) is the intensity of corresponding pixel in the smoothed image, and \( T_g \) is a specific threshold.

Taking into account the different texture and illumination, the contrast between crack pixels and its surrounding background varies significantly, so the threshold \( T_g \) is adaptively determined. To decrease the effect of inhomogenous texture and illumination, we divide the whole image into grid cells, denoted as \( C_i, i = 1, \ldots, n_c \). In each grid cell \( C_i \), we select the threshold \( T_k \) as

\[
T_k = \mu_i + k_g \sigma_i,
\]

where \( \mu_i \) and \( \sigma_i \) are the mean and standard deviation of the difference between original intensities and their corresponding smoothed intensities of all pixels within \( C_i \), \( k_g \) is an adjustment parameter which we generally set to be 1 in our application.

Since the image of airport runway may quite noisy, as illustrated in Fig. 3. The extracted candidate crack pixels based on intensity property contains many isolated or non-crack ones, which need to be found and removed according to their shape characteristic, as presented in the following step.

2) SHAPE BASED CRACK PIXEL FILTERING

The shape characteristic is used to distinguish the cracks from other dark patches in the image. First, we classify all candidate crack pixels into different groups, denoted as \( G_i, i = 1, \ldots, g_n \), according to their connectivity. To remove the noisy smaller blobs, we only keep the groups which contain more than \( T_b \) pixels and whose contour/area ratio is large enough.

\[
|G_i| \geq T_b, \quad \frac{c(G_i)}{a(G_i)} > T_r, \tag{7}
\]

where notation \( | \cdot | \) means the size of a collection, \( c(G_i) \) and \( a(G_i) \) denote the contour length and area of \( G_i \), respectively, \( T_r \) is a specific threshold.

While by applying the above process most crack pixels are detected, noises are also brought in, specifically linear pavement textures for concrete pavement, which possess the similar property, as shown in Fig. 3. However, pavement textures always appear as parallel straight lines, which makes them different from cracks. Thus, firstly, isolated pavement texture is detected as follows. At one hand, a minimum bounding box is detected for each candidate crack pixel group \( G_i \), that has a narrow width can be inferred as containing a straight line, which can be classified as pavement texture. At the other hand, pavement texture that is connected to cracks is detected as follows. Skeleton extraction based on K3M algorithm [27] is applied for each connected component \( G_i \) to extract center axis of the patch, after which line detection using LSD [28] line segment detector is applied, and parallel straight lines can be classified as pavement textures. Morphological processing is applied afterwards to the detected pavement textures, taking into account that the pavement textures are always several pixels wide.

Leaving out the pavement texture pixels from the candidate crack pixels, we obtain the skeleton of cracks. Then we perform crack region growing by absorbing dark pixels neighboring the currently detected cracks, if the intensities of these pixels can satisfy a threshold test. The intensity threshold \( T_n \) is determined as follows,

\[
T_n = \mu_c + k_c \sigma_c, \tag{8}
\]

where \( \mu_c \) and \( \sigma_c \) are the mean and standard deviation of the grey levels of all currently detected crack pixels, and the parameter \( k_c \) is set to be 1 in our experiments. This aggregation process is iteratively performed, so that all potential crack pixels surrounding the crack skeleton can be incorporated.

Our proposed crack detection method is summarized in Algorithm 2.
Algorithm 2 Crack Detection Algorithm

\begin{verbatim}
input : Image I
output: all crack pixels in I

1. Divide I into grid cells Ci, i = 1, \ldots, n_c;

2. foreach Ci do
   3. Compute the intensity threshold T_g using (6);
   4. Extract candidate crack pixels using (5);
   5. Group candidate crack pixels to form Gi, i = 1, \ldots, n_g;

6. foreach Gi do
   7. if (7) is satisfied then
       8. Remove Gi;
       9. Remove Gi if it belongs to a group of parallel line segments;
      10. Extract the skeleton of Gi using K3M algorithm;
      11. Iteratively perform crack growing by absorbing dark pixels neighboring to Gi if their intensities satisfy (8);

12. return all crack pixels;
\end{verbatim}

C. SUBSURFACE DEFECT DETECTION FROM GPR

As we discussed in Section II, one of the main reasons, that existing CNN-based method cannot be directly deployed in applications, is that only 2D B-scan images are employed, which ignores the natural 3D property of subsurface defects. Thus, we propose an algorithm to handle this problem.

Before introducing our algorithm, we first give some definitions and notations. While the GPR scans on the ground along a linear trajectory, it generates a B-scan image, that records the reflected pulses and shows the subsurface situation. When the GPR fixed on robot moves over a regular grid to collect multiple parallel B-scans, it produces a series of B-scans. This ensemble of B-scans forms a 3D data which is named as a C-scan. Denote B_k as the k-th B-scan image, C = \{B_k\} as the C-scan consisting of n_b B-scans. We can evenly divide the 3D C-scan into slices of 2D images from main view, top view, and left view, respectively. The 2D images from main view are B-scan images. We denote T_m, m = 1, \ldots, n_m and L_n, n = 1, \ldots, n_l as the images sliced from top view and side view, respectively.

We propose a voting-based strategy to fuse the information of 2D images sliced from all views of C. We first adopt a CNN-based 2D object detection algorithm, such as Faster R-CNN, to generate the 2D bounding boxes labeled with categories and probabilities as the candidate defect regions. Denote r_j as the j-th bounding box, with its probability belonging to category c_k as p_j,k. For each point X_i in C, we define its score belonging to category c_k to be the sum of probabilities from all 2D bounding boxes of B_k, T_m and L_n. Thus, we compute the score as

\begin{equation}
s(X_i, c_k) = \sum_{j} \kappa_{i,j} p_{j,k},
\end{equation}

with

\begin{equation}
\kappa_{i,j} = \begin{cases} 1, & \text{if } X_i \in r_j \\ 0, & \text{otherwise} \end{cases}
\end{equation}

where X_i \in r_j indicates point X_i being inside of r_j.

With the scores of all points in C computed, we only keep the points if the following criteria is satisfied.

\begin{equation}
s(X_i, c_k) > T_s,
\end{equation}

where T_s is a specific threshold for point scores.

Then, we cluster all remained points according to their positions and residing categories. DBSCAN algorithm [29] is adopted for this clustering step. DBSCAN starts with a remained point X_i of category c_k, and retrieves all points of the same category density-reachable from X_i with regard to two parameters, \(\varepsilon\) and MinPts. The \(\varepsilon\)-neighborhood of X_i is defined as,

\begin{equation}
N_\varepsilon(X_i) = \{X_j \mid \|X_i - X_j\| \leq \varepsilon, \forall X_j \text{ belongs to } c_k\}.
\end{equation}

where \(N_\varepsilon(X_i)\) contains at least \(\text{MinPts}\) of remained points. In our experiments, we set \(\varepsilon = 16\) and \(\text{MinPts} = 4\).

Finally, for each point cluster, we generate a 3D bounding box as the detected defect region.

We summarize our subsurface defect detection method in Algorithm 3.

Algorithm 3 Subsurface Defect Detection Algorithm

\begin{verbatim}
input : C
output: 3D bounding boxes of defects

1. Divide C into a set of B_k, T_m and L_n;
2. Generate candidate 2D bounding boxes from B_k, T_m and L_n using Faster R-CNN;
3. foreach X_i do
   4. Compute its score s(X_i, c_k) using (9);
   5. Filter all points in C using (11);
   6. Cluster the remain points according to their spacial intensities and categories using (12);
   7. Generate 3D bounding box for each point cluster;
8. return all 3D bounding boxes;
\end{verbatim}

V. EXPERIMENTS

We have employed our ARIR system in over 20 airports of China. To evaluate our proposed algorithms thoroughly, we select a representative airport of southeast of China for both quantitative and qualitative analysis. The surveyed region is over 2000 m^2, and subsurface depth is 1.53 meters. Two human experts have labelled the surface cracks and subsurface defects individually. With the collected inspection data, we fulfill to perform both the surface and subsurface defect detection and visualization.
FIGURE 4. Large scale image stitching results of a region of 20m × 405m, with 11642 original images. We resize the scale between horizontal and vertical directions for a better visualization effects.

TABLE 1. Surface crack detection results.

| Method               | Precision | Recall | F1-measure |
|----------------------|-----------|--------|------------|
| Our crack detection method | 66%       | 80%    | 70%        |

A. LARGE SCALE IMAGE STITCHING RESULTS
We succeed to stitch the large amount of images of the airport runways. An example of image stitching results of a region of 20m × 405m, with a total of 11642 original images is shown in Fig. 4, where we can see that the panorama of the surveyed airport runway is generated.

B. SURFACE CRACK DETECTION RESULTS
To quantitatively evaluate the performance of our crack detection method, we select the representative images to form a dataset, consisting of 112 images with the resolution of 1200 × 900 pixels. We employ three well-known metrics, including precision, recall and F1-measure, for the evaluation. The ground truth is manually labelled by human carefully. Considering it is inevitable to introduce errors when labelling, we assume that true positive pixels are included within a 5-pixel vicinity of the ground truth. These three metrics can be computed based on true positive (TP), false negative (FN), and false positive (FP),

\[
\text{Precision} = \frac{TP}{TP + FP} \quad (13)
\]

\[
\text{Recall} = \frac{TP}{TP + FN} \quad (14)
\]

\[
\text{F1-measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (15)
\]

We present some example crack detection results in Fig. 5, where we can see that the images are quite challenging due to the special texture, the thin width of cracks and poor lighting conditions.

The statistical quantitatively results are shown in Tab. 1. The results of our proposed algorithm can satisfy the requirement of field applications.

C. SUBSURFACE DEFECT DETECTION RESULTS
We compare our algorithm with Faster R-CNN which is only conducted on B-scan. To evaluate the performance of different methods quantitatively, three metrics, including Precision, Recall and F1-measure, are employed. To illustrate these metrics for object-level detection, we employ the indicator IoU which represents the overlapping rate of overlap between the resulting box and the labelled box. If the IoU value is greater than a specific threshold \( T_{\text{IoU}} \), we consider this box as a true positive. The quantitative results are shown in Tab. 2, where we can see that our proposed algorithm achieves a better performance compared with standard Faster R-CNN.

Fig. 6 presents an example of defect qualitative results from two different views of a GPR C-scan segment. This global subsurface defect map can be used for quantitative analysis and also for visually assessing global defect patterns.
developed. We tested our proposed system in applications. A data analysis software embedding these algorithms was an acceptable global image of the large scale airport runway. The image stitching algorithm allowed to generate data analysis was presented to perform the subsurface defect detection. The image stitching algorithm allowed to generate an acceptable global image of the large scale airport runway. A data analysis software embedding these algorithms was developed. We tested our proposed system in applications over 20 airports and achieved satisfying performance.

VI. CONCLUSION
We developed the automated airport runway inspection data collection and analysis. Specifically, we proposed a crack detection method to detect the surface cracks robustly from the collected images. A deep learning algorithm for GPR data analysis was presented to perform the subsurface defect detection. The image stitching algorithm allowed to generate an acceptable global image of the large scale airport runway. A data analysis software embedding these algorithms was developed. We tested our proposed system in applications over 20 airports and achieved satisfying performance.

REFERENCES
[1] H. M. La, R. S. Lim, B. B. Basily, N. Gucunski, J. Yi, A. Maher, F. A. Romero, and H. Parvardeh, “Mechatronic systems design for an autonomous robotic system for high-efficiency bridge deck inspection and evaluation,” IEEE/ASME Trans. Mechatronics, vol. 18, no. 6, pp. 1655–1664, Dec. 2013.
[2] H. M. La, N. Gucunski, K. Dana, and S.-H. Kee, “Development of an autonomous bridge deck inspection robotic system,” J. Field Robot., vol. 34, no. 8, pp. 1489–1504, Dec. 2017.
[3] S.-N. Yu, J.-H. Kang, and C.-S. Han, “Auto inspection system using a mobile robot for detecting concrete cracks in a tunnel,” Autom. Construct., vol. 16, no. 3, pp. 253–261, May 2007.
[4] M. R. Jahnanshahi, S. F. Masi, and G. S. Sukhatme, “Multi-image stitching and scene reconstruction for evaluating defect evolution in structures,” Struct. Health Monit., Int. J., vol. 10, no. 6, pp. 643–657, Nov. 2011.
[5] J. Jia and C.-K. Tang, “Image stitching using structure deformation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 30, no. 4, pp. 617–631, Apr. 2008.
[6] J. Zaragoza, T. J. Chin, Q. H. Tran, M. S. Brown, and D. Suter, “As-projective-as-possible image stitching with moving DIT,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2013, pp. 2339–2346.
[7] M. Brown and D. G. Lowe, “Automatic panoramic image stitching using invariant features,” Int. J. Comput. Vis., vol. 74, no. 1, pp. 59–73, Apr. 2007.
[8] L. Juan and G. Oubong, “SURF applied in panorama image stitching,” in Proc. 2nd Int. Conf. Image Process. Theory, Tools Appl., Jul. 2010, pp. 495–499.
[9] J. Tang and Y. Gu, “Automatic crack detection and segmentation using a hybrid algorithm for road distress analysis,” in Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC), Sep. 2013, pp. 3026–3030.

### TABLE 2. Subsurface defect detection results.

| Method                  | Precision | Recall | F1-measure |
|-------------------------|-----------|--------|------------|
| Faster R-CNN            | 50%       | 55%    | 53%        |
| Our defect detection method | 61%       | 73%    | 67%        |

FIGURE 6. An example of subsurface defect qualitative results from top view (top) and side view (bottom) of a GPR C-scan segment. The same number indicates the same defects.
ZHONGCHENG GUI was born in Lu’an, China, in 1979. He received the B.S. and M.S. degrees in mechanical engineering from Tongji University, Shanghai, China, in 2006, and the Ph.D. degree in robotics from Tsinghua University, Beijing, China, in 2010. He is the CEO of Chengdu Guimu Robot Company, Ltd., Chengdu, China. He has authored or coauthored over 20 technical articles and holds over 100 patents. His research interests include mechanics, and robot localization and navigation.

HAIFENG LI was born in Tongliao, China, in 1984. He received the B.S. degree in computer science and technology and the Ph.D. degree in control theory and control engineering from Nankai University, Tianjin, China, in 2007 and 2012, respectively.

He is an Associate Professor with the Department of Computer Science and Technology, Civil Aviation University of China, Tianjin. He has authored or coauthored over 30 technical articles. His research interests include computer vision, image processing, robotic sensing, multisensor fusion, and robot localization and navigation.

* * *