IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 15, 2021, accepted October 10, 2021, date of publication October 14, 2021, date of current version November 9, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3120207

A Coarse-to-Fine Approach for Rock Bolt Detection From 3D Point Clouds

SARP SAYDAM[1,2], BOGE LIU[2], BINGHAO LI[2], (Member, IEEE), WENJIE ZHANG[3], SARVESH KUMAR SINGH[2], AND SIMIT RAVAL[2]

[1]Digital Development Engineer, DSI Underground Group, Newcastle, NSW 2290, Australia
[2]School of Minerals and Energy Resources Engineering, UNSW Sydney, Sydney, NSW 2052, Australia
[3]School of Computer Science and Engineering, UNSW Sydney, Sydney, NSW 2052, Australia

Corresponding author: Boge Liu (boge.liu@usnw.edu.au)

ABSTRACT Rock bolts have been widely used to enhance the structural stability of underground infrastructures. Careful tracking of rock bolt positions is highly significant since it assists with operational success of ground support and has applications to predictive maintenance practices. This paper presents a practical algorithm, `CFBolt`, to detect rock bolts from a 3D laser scanned point cloud. Considering that rock bolts are relatively tiny objects, `CFBolt` follows a two-step coarse-to-fine strategy. It first computes a single-scale proportion of variance (POV) for each point as the local point descriptor and filters out near 95% not-bolt points with a simple but effective classifier, Linear Discriminant Analysis (LDA), which allows for the pruned point cloud to be then used as a compatible input to a deep neural network, designed and trained to precisely detect rock bolts from the pruned point cloud. `CFBolt` was tested for detecting rock bolts from LiDAR scan data collected from Sydney's civil tunnelling project site. The entire dataset contains more than 160 million points. The obtained scores of Intersection over Union (IoU) and precision for individual bolt points were 89.33% and 92.04%, respectively. For rock bolt objects, the precision and recall were 98.34% and 98.73%, respectively. The detection quality of `CFBolt` is superior to the state-of-the-art 3D object detection algorithms and the newest rock bolt detection algorithm, demonstrating the robustness and effectiveness of `CFBolt`.

INDEX TERMS Rock bolt, point cloud, LiDAR, neural network.

I. INTRODUCTION

Rock reinforcement, which refers to the support that is intrusive to the rock mass, such as rock bolts [26], is an essential part of every tunnelling project as it consolidates the stability of the tunnel structure. Rock bolts are steel rods that are installed into a rock mass to provide geotechnical support and are an essential part of any underground operation [17], [21]. Rock bolts can meet a variety of geological conditions and requirements necessary for a modern ground support system in an underground mine or tunnel [23]. The purpose of rock bolts in underground environments can be summarised as serving two essential functions: suspending potentially loosened blocks together and forming a protective pressure arch around the excavated void to help the ground strengthen itself [20].

Careful tracking of rock bolt positions is highly significant since it can assist with operational success of ground support and has applications to predictive maintenance practices [6], [24], which can highlight problems with bolting pattern design and demonstrate areas with poor rock conditions that may have been overlooked. However, it is challenging to monitor rock bolts in-situ due to the limitation of lighting and unavailability of global navigation signals in the underground environment [32], [35].

Traditionally, deformation and support conditions in underground tunnels have been monitored through visual inspection [25] and geotechnical instrumentation [11], [31]. However, the subjectivity of visual observation techniques can result in ambiguous or incomplete analyses with little quantifiable data [3]. Monitoring displacements with conventional instrumentation can be expensive and time-consuming, and the information collected is typically limited to just a few locations [3], [7]. Other popular surface imaging approaches such as photogrammetry are subject to the poor lighting condition of the environment [33]. [18], [39] have recently studied the 2D image-based deep learning algorithms, such as mask R-CNN, and demonstrated its effectiveness for

The associate editor coordinating the review of this manuscript and approving it for publication was Yiming Tang.

monitoring deformation and support conditions in tunnels, which shows the potential of deep learning algorithms for such tasks.

In recent years, LiDAR, a technique used in laser scanning for 3D mapping, has been widely used for underground 3D imaging [36]–[38]. In underground environment, laser scanning has normally been used to monitor tunnel closures [9], deformations [10], [14], [40], and cavity collapse [8]. With a large scanning range and being independent of lighting condition [13], LiDAR offers new opportunities for underground ground support monitoring, in particular, rock bolt identification. The output of LiDAR forms a set of points with XYZ coordinates, which is called a point cloud. The point cloud represents an exact 3D replica of the scanned environment.

Identifying objects from point clouds has been widely studied in the literature. Numerous deep learning algorithms have been proposed, such as PointNet [27], PointNet++ [28], and SO-Net [19]. Rahman *et al.* [29] provides a comprehensive survey on 3D object detection with deep neural networks. Although these approaches achieve impressive results for 3D object identification, almost all of them are limited to tiny 3D point clouds or fixed input sizes (e.g., 4k points or 1 × 1 meter blocks) and cannot be directly extended to larger point clouds [15]. Recently, there have been several works focusing on extending deep learning algorithms to large point clouds, such as RandLA [15] and Cylinder3D [41]. The methods downsample the input cloud and keep as much local structural information on the remaining points as possible by aggregating local features on sampled points. However, they are not suitable for identifying rock bolts. As shown in Figure 1, one tunnel scan consists of around 2 million points while one rock bolt typically accounts for 300 points. The relatively small size of rock bolts makes it hard for existing algorithms to be directly applied to detect rock bolts. The main reason is that few points from rock bolts will be preserved after the downsampling process. For example, if the downsampling rate is 1%, the expected number of points for one rock bolt is only 5 after the downsampling process. Even if the sampling possibility is not uniform, the remaining number of points from rock bolts would still be too small to represent the bolt structure, and the bolt features can be easily lost.

A. ROCK BOLT DETECTION ALGORITHM

Due to the critical role of rock bolts in reinforcement, many works study the detection of rock bolts based on LiDAR scan. Martínez-Sánchez *et al.* [22] used rock bolts as anchors to computed rigid transformation between an initial and posterior point cloud, which is called registration. The thickness of sprayed concrete is then calculated based on the registration result. They proposed a deep network that consists of two autoencoders and an output layer to detect rock bolts. However, the detection result contains many false positives as the algorithm for rock bolt detection only serves as a pre-processing step for point cloud registration, and it focuses on finding as many rock bolts as possible (achieving a high recall rate) while ignoring the precision.

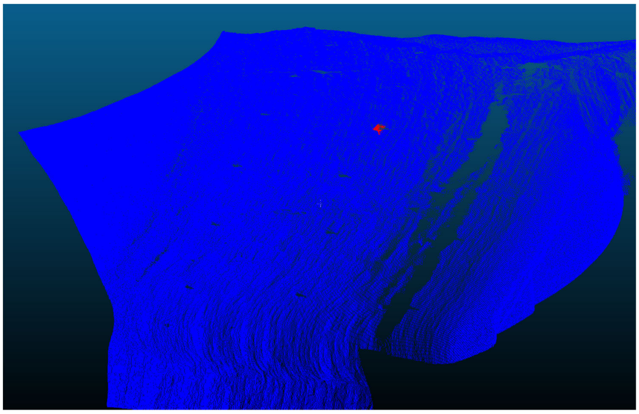


FIGURE 1. A scan of underground tunnel. Red points consist of a rock bolt. The entire scan usually contains 2 million points, while a single bolt typically contains 300 points.

Gallwey *et al.* [12] proposed a machine learning approach to detect rock bolts. They construct a 65 dimensional feature vector for each point, which includes eigenvalue based neighborhood features [4], density, and fast point feature histogram features (FPFH) [30]. Several machine learning (ML) classifiers are tested to label the point cloud based on the feature vector. Among them, multilayer perceptron and random forests are finally chosen. Though the ML classifiers are carefully selected, the point-wise feature vector is user-specified and not learnt by the deep neural network, limiting its representation power. Compared with the F1scores (the harmonic mean of the precision and recall) of point-wise classification and rock bolt detection reported in their paper, which are 0.64 and 0.90, our proposed algorithm improves the score by 38% and 9% to 0.88 and 0.98, respectively.

Singh *et al.* [34] proposes the newest rock bolt detection algorithm (we call this algorithm LDANN in the rest of this paper). Similar to [12], LDANN generates a local descriptor for each point using eigenvalue based feature and FPFH. Then a single-layer neural network is trained for bolt classification with the local descriptor. Moreover, as shown in Figure 2, LDANN only recognises a circular region over bearing plates instead of square and determines whether there exists a bolt or not based on the detected region. The bearing plates with a structure similar to the tunnel surface (green points) are hard to detect using local point descriptors. Therefore, it is challenging to acquire the exact shape and boundary of rock bolts using LDANN, which affects future possibilities in identifying changes in bearing plates.

This paper presents an effective and efficient method, named `CFBolt`, to detect rock bolts from point clouds. To avoid the manually pre-processing step and achieve fully automated rock bolt detection, `CFBolt` follows a two-step coarse-to-fine manner. `CFBolt` first filters out most non-bolt points and limits the search space to a few locations. Then it extracts points from these locations within a fixed radius

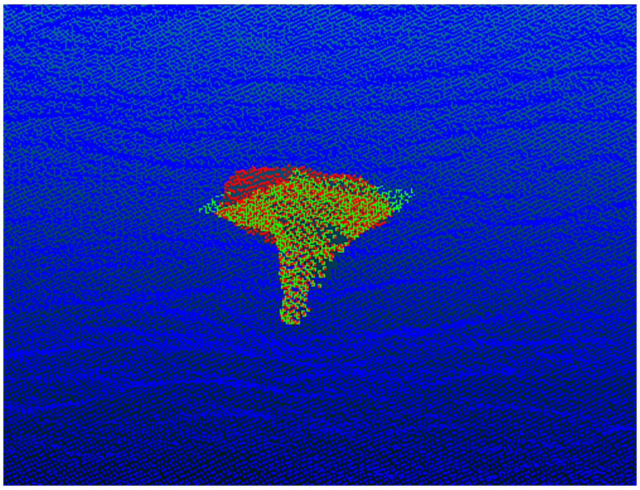


FIGURE 2. LDANN misses the exact boundary of bearing plates (shown in green) and identifies only a circular region (shown in red) over it.

and feeds them to a deep neural network to detect rock bolts. To achieve robust and high-quality bolt detection results, we use the deep neural network to learn a global feature vector and local feature vectors and combine them to generate per point feature vector. A transformer net is adopted to ensure that the learned feature vector is consistent under certain geometric transformations. As shown in Section IV, `CFBolt` achieves high-quality results and makes an improvement compared to existing algorithms, such as 12% improvement over LDANN on point-wise classification.

II. DATASET

In order to train the neural network, a large amount of labelled data is required. As no existing dataset of labelled rock bolts is available, we used a tripod-mounted terrestrial laser scanner to collect rock bolt data from a civil tunnelling project site in Sydney, New South Wales, Australia. The scanning was conducted in conjunction with the excavation cycle of road-header excavation, rock bolt installation followed by shotcrete application. The collected data consists of 84 separate scans, including the bolted rock face of excavation cuts ranging from 2–6 metres. All the 84 scans are similar in terms of geometry, bolt density, and scale. Each scan was taken after rock bolt installation and before the surface was covered with shotcrete. The scanner was positioned 1-3m away from the excavated section and covered a 180-degree area facing the heading. The number of bolts captured in each scan ranged from 10–20, at a resolution of 6.3mm, which is the Euclidian distance between points at 10 metres from the scanner. This amounted to a total of 1,266 rock bolts in the dataset. The rest of the data included the surrounding rock (mainly Sydney Basin Hawkesbury Sandstone), as well as areas of previously applied shotcrete, ventilation, services, and equipment.

The rock bolt points are classified manually and given the class label '1' while all the other not-bolt points are labelled '0'. No manual cleaning method is applied to remove not-bolt points from the tunnel surface, and the final dataset represents the true underground situation. Table 1 provides the statistics of the dataset in this study.

TABLE 1. Statistics of the dataset used in our experiments.

# of points	# of bolt points	# of bolts
165M	347k	1,266
avg # of points / scan	avg # of points / bolt	avg # of bolts / scan
1.96M	274	15

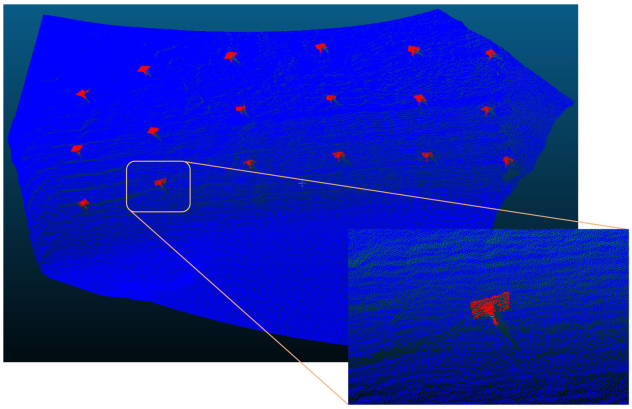


FIGURE 3. One of the scans used in our research. The scan consists of 1.86 million points and contains 18 rock bolts. The bolt shown in the zoom-in figure consists of 372 points.

Figure 3 gives one of the scans used in our research. It consists of 1,859,536 points. There are 18 rock bolts in the scan, which consists of 6,941 points in total. The rock bolt shown in the zoom-in figure at the left corner contains 372 points.[1]

III. METHODOLOGY

One of the superiority of using the deep neural network for 3D object detection is that the neural network itself learns the object's structural features. However, the neural network used in existing bolts detection methods [12], [33] are small, typically consists of one or two hidden layers and are only used for classification with user-specified per point feature vector, such as the proportion of variance (POV) [5], and FPFH. A drawback of using a user-specified feature vector is limiting the neural network's generalisation ability to fit unseen data. To overcome this, a straightforward solution is to increase the dimension of feature vectors. For example, Gallwey *et al.* [12] uses a 65-dimensional feature vector. Though this improves the performance for rock bolt detection, the time cost of constructing a high dimensional feature vector increases, and the limitation of generalisation ability still exists.

[1] More images about the dataset are available at https://github.com/brandonliu0306/tunnel_images

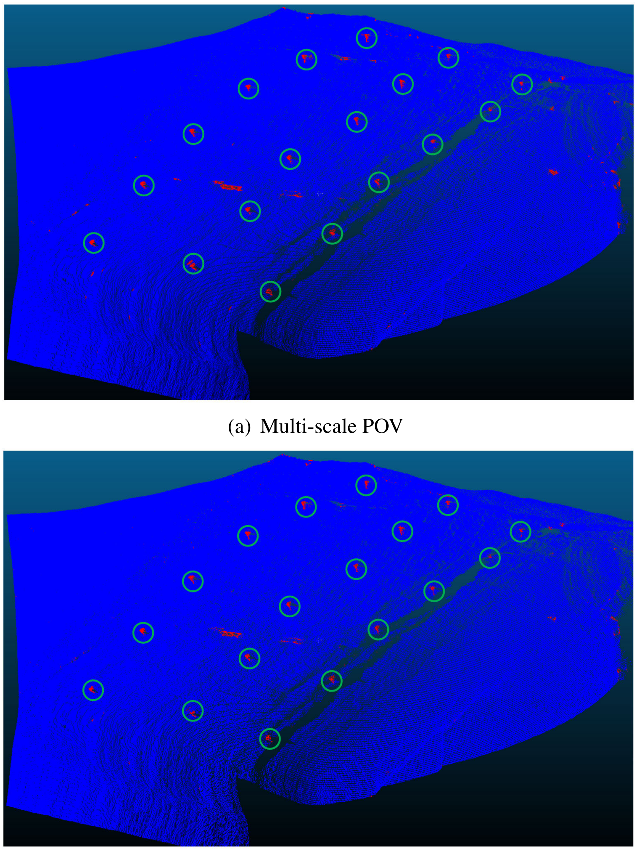
To identify rock bolts, our proposed rock bolt detection method, `CFBolt`, adopts a two-step coarse-to-fine strategy. The first step, called Rough Estimation, estimates possible locations of rock bolts while pruning as many not-bolt points as possible. Then, a set of candidate bolts is generated by extracting a 0.2m radius sphere centred at each location as the extruding portion of rock bolts is assumed to be no longer than 0.3m in length. The second step, Precise Detection, adopts an effective rock bolt detection neural network to classify and segment rock bolts from candidate bolts.

A. ROUGH ESTIMATION

Rough Estimation generates a set of candidate rock bolts $\mathcal{C}$ in which each may or may not contain a real rock bolt. The candidate rock bolts will be further classified at the second step. Rough Estimation needs to prune out as many not-bolt points as possible while ensuring no rock bolt will be filtered out. To this end, we use the single-scale POV [5] as per point structural descriptor. Note that using more structural descriptors, like FHFP as shown in [12] can filter out more not-bolt points, but some real rock bolts may also be pruned. Also, it is efficient to compute one simple structural descriptor for each point as the original scan typically contains about 2 million points.

POV defines the appearance of a point cloud, whether linear, planar or 3D surface at a given scale based on the proportions of eigenvalues. Given a point of interest s_i, for points in the neighbouring ball centred on s_i, three eigenvalues are calculated from Principal Component Analysis (PCA) performed on the recentered coordinates of the points in that ball. The diameter of the neighbouring ball is controlled by the user-specified scale. Let λ_1, λ_2, and λ_3 be the three eigenvalues results from the PCA. POV is described as the normalized eigenvalue, i.e., $p_i = \frac{\lambda_i}{\lambda_1+\lambda_2+\lambda_3}$. Given the constraint $p_1 + p_2 + p_3 = 1$, POV is defined as a two-parameter feature at any given point and scale, i.e, (p_1, p_2).

For each point, a single-scale POV is computed at a 10cm scale as performing PCA transform is time-consuming, and the rough estimation step does not require very high precision. The 10cm scale is selected by following [34] which suggests that POV is efficient in capturing the geometrical features of rock bolts up to 10 cm in length. Section IV-A further shows that 10 cm scale is a good choice for single-scale POV (Figure 9) because no real rock bolt will be wrongly filtered out. Note that in [5], [33], multi-scale POV is computed at different scales ranging from 2cm to 10cm with 1cm step. Though multi-scale POV can capture more local structure details, as shown in Figure 4, multi-scale POV does not perform better than single-scale POV for identifying locations of rock bolts. This is because a rough estimation of rock bolt locations does not require very detailed structural information. Furthermore, the number of false-positive points (red points outside the green circles) in Figure 4 (a) is more than that in Figure 4 (b), which shows that, as proved in our experiment (Section IV-A), too much detailed structural information may even hinder filtering out not-bolt points. By considering both the efficiency and effectiveness, single-scale POV at a 10cm scale is chosen in this study.

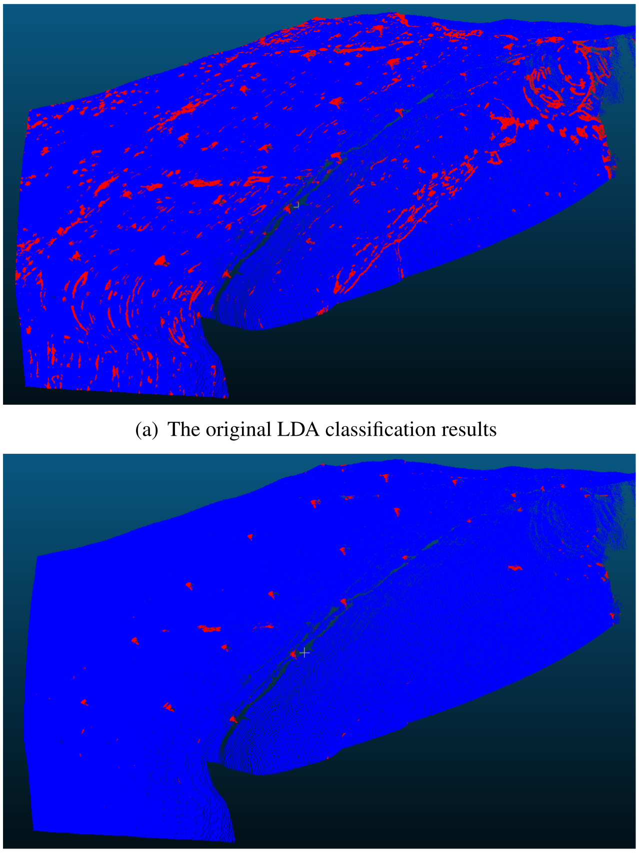

(a) Multi-scale POV

(b) Single-scale POV

FIGURE 4. The detection results using Linear Discriminant Analysis (LDA) with multi-scale POV and single-scale POV. Both find locations of all the rock bolts (green circles). But the computation cost of single-scale POV is only a ninth of multi-scale POV. In figure (a), multi-scale POV is computed at scale ranging from 2cm to 10cm with an increment of 1cm. In figure (b), single-scale POV is computed at scale of 10cm.

A simple linear classifier, Linear Discriminant Analysis (LDA) [2], is adopted for classifying points based on the POV descriptor computed in the previous step. The principle behind LDA is to project the samples in high dimensional feature space onto a line such that samples from the same class are as close to each other as possible while samples from different classes are as far away from each other as possible. LDA proposes to set the line (vector) as $(\Sigma_0 + \Sigma_1)^{-1}(\mu_1 - \mu_0)$, where Σ_c and μ_c are the covariance matrix and the mean vector of the samples in class c (not-bolt and bolt). In other words, a point's label is determined on a certain side of a hyperplane perpendicular to this line. We use the normalised distance to the hyperplane as the confidence for the classification of each point. The further a point is from the hyperplane in the feature space, the higher probability it has belonging to that class.

A confidence threshold is used to prune not-bolt points further. For any point classified as bolt point in LDA, if its confidence is no larger than 0.9, we still treat it as a not-bolt point. As shown in Figure 5, by applying the threshold to LDA classification results, the number of false-positive results are significantly reduced. The bolt points are concentrated on the spike part of the rock bolts, which is the most significant feature of the rock bolt compared with the tunnel surface. Though the shapes of rock bolts are not detected completely, the detected bolt points are enough to estimate rock bolts' locations.

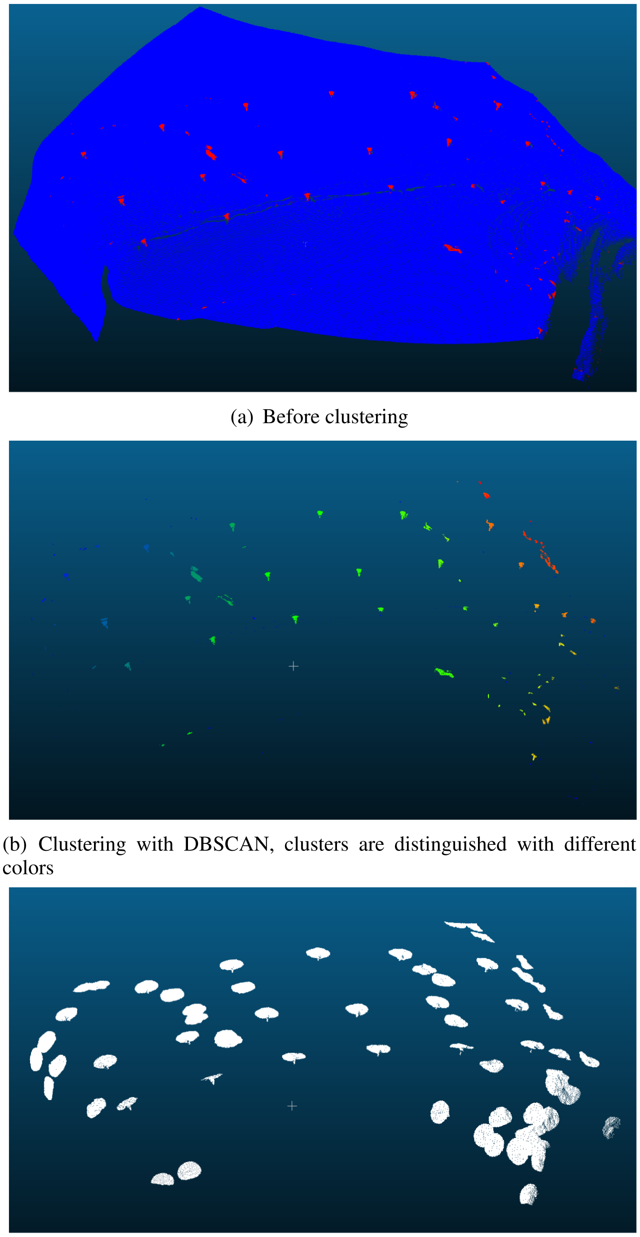


(a) The original LDA classification results

(b) After pruned with confidence threshold

FIGURE 5. The classification result before and after pruned with confidence threshold. Most false positive results are filtered out.

We apply density-based spatial clustering of applications with noise (DBSCAN) to the detected bolt point in the previous step. DBSCAN generates clusters by searching core points and grouping points within the distance threshold ϵ to them. Considering the point cloud density and the size of rock bolts, the maximum neighbourhood distance ϵ is set to 5cm, and the minimum cluster size is set to 10. A cluster's central coordinate is computed as the mean value of coordinates of all the points belonging to that cluster. All the points within a 0.2m radius sphere centred at the central coordinate of a cluster will be extracted as a candidate rock bolt. Figure 6 demonstrates the results of performing DBSCAN on the LDA classification. Candidate rock bolts are generated based on the central location of each cluster.



(a) Before clustering

(b) Clustering with DBSCAN, clusters are distinguished with different colors

(c) Candidate rock bolts

FIGURE 6. After the LDA result is acquired (a), DBSCAN is performed to generate clusters (b). Then candidate bolts are extracted as a 0.2 radius sphere centred at each cluster (c).

B. PRECISE DETECTION

Currently, deep learning is the dominant method for recognising objects from three-dimensional data. It has been shown that deep neural networks can arbitrarily approximate any continuous set function given enough neurons [27], which means it has the potential to detect any object shape. Compared with existing bolt detection algorithms, the most characteristic part of deep learning algorithms is that they learn

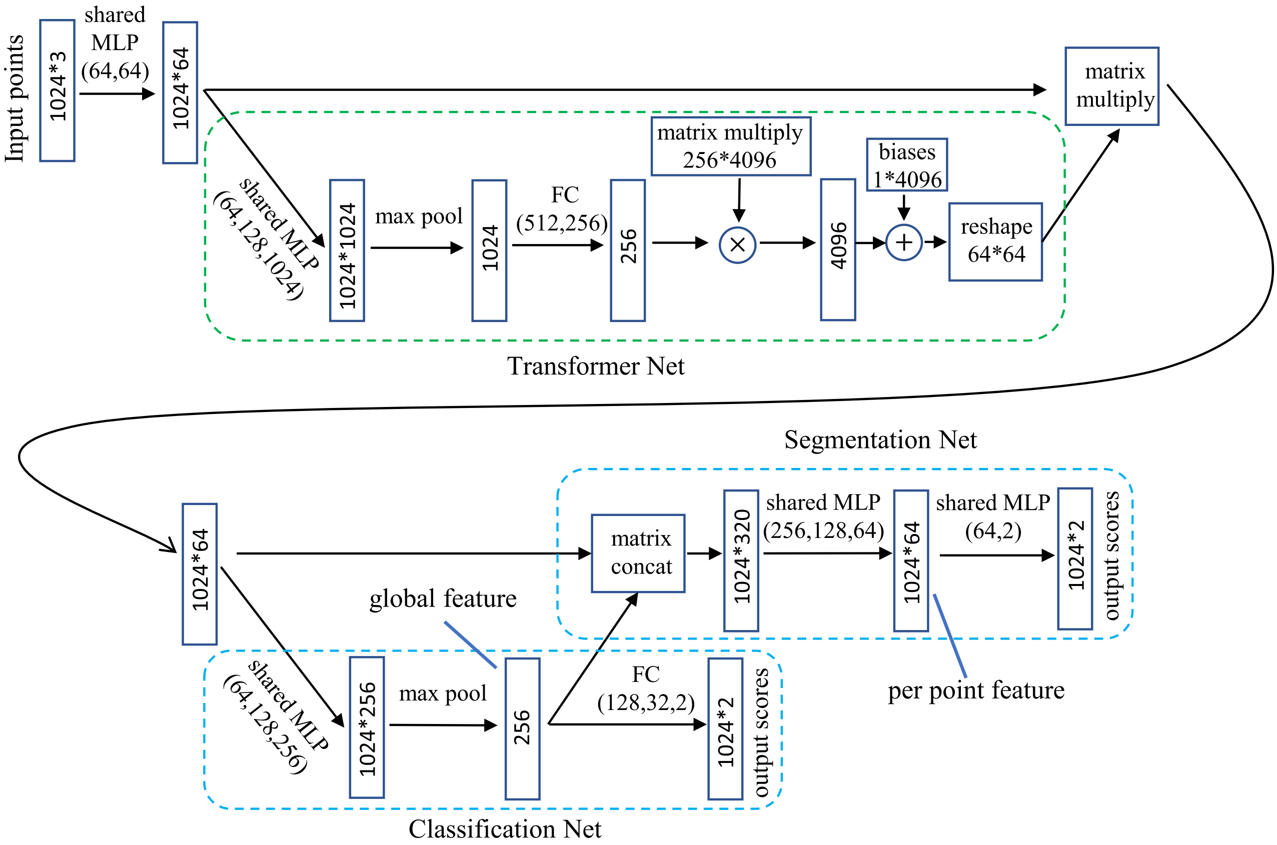


FIGURE 7. RBNN architecture. "MLP" and "FC" stands for multi-layer perceptron and fully-connected layers, respectively. Numbers in bracket are layer sizes. Batchnorm is used for all layers with Rectified Linear Unit (ReLU).

per point features instead of using user-specified feature descriptors. Therefore, it has the capacity to generate more accurate features representing rock bolts given the proper training data.

The structure of our rock bolt detection neural network (RBNN), which is based on the pioneering work PointNet [27], is visualised in Figure 7. To elaborate RBNN structure, we divide it into three sub-networks, e.g., Transformer Net, Segmentation Net, and Classification Net. Transformer Net provides pose normalisation for the input candidate rock bolts to ensure the output results are invariant to certain geometric transformations. Classification Net determines whether a real rock bolt exists in the input, and Segmentation Net outputs per point labels to detect the exact shape of rock bolts.

The input layer accepts a fixed number of points. Note that the number of points in candidate rock bolts is not fixed because the point could density varies depending on the distance from the scanner and the shape of captured objects. In this study, the input point to RBNN is set to 1024 as the mean value of the number of points per candidate rock bolt is 993. Before the candidate rock bolt is fed to RBNN, we resize it to 1024 by randomly duplicating or removing the points in it. If some point is removed, its label will be determined by the majority vote from its top 9 nearest neighbours. For each candidate rock bolt, we recenter and rescale its points' coordinates to fit in a unit cube before feeding it to RBNN and map them back to original coordinates after classification or segmentation. Suppose that the maximum and the minimum value among all the points' coordinates in a candidate rock bolt are $x_{min}, y_{min}, z_{min}$ and $x_{max}, y_{max}, z_{min}$, respectively, the normalized coordinate for point (x, y, z) will be

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{bmatrix} \frac{1}{x_{max}-x_{min}} & & \\ & \frac{1}{y_{max}-y_{min}} & \\ & & \frac{1}{z_{max}-z_{min}} \end{bmatrix} \begin{pmatrix} x-x_{min} \\ y-y_{min} \\ z-z_{min} \end{pmatrix}$$

RBNN first uses a shared multi-layer perceptron (MLP) to map every 1024 points from three dimensions (x, y, z) to 64 dimensions. Note that for shared MLP, each of the 1024 points uses the same multi-layer perceptron, and the mapping is identical and independent on each point. The classification and segmentation of rock bolts should be invariant to certain geometric transformations, such as rotation. Motivated by Spatial Transformer Networks [16], RBNN adopts a Transformer Net sub-network to seek to provide pose normalisation for a given input. The Transformer Net predicts an input-dependent transformation matrix that is then matrix multiplied with the input to achieve an appropriate rigid transformation. Because each of the 1024 input points is represented as a vector and are mapped to the 64-dimensional embedding space independently, applying a rigid transformation simply amounts to matrix multiplying each point with a 64 × 64 transformation matrix A. Assuming for point p_i, its input corresponding 64-dimensional vector is v_i, the output of the rigid transformation v_i' will be

$$v_i' = Av_i$$

As each point's coordinate has been normalised before feeding into RBNN, the translation item in the above transformation is omitted. During the training process, we use $L = ||I - AA^T||^2$ regularisation term to constrain the resulting 64-by-64 transformation matrix (represented as A) to approximate an orthogonal transformation.

As shown in Figure 7, the Transformer Net first uses shared MLP to further expand the dimensions to 1024. A max pool layer is used to encode a global feature vector with 1024 dimensions. The dimensionality is reduced to 256 by two fully connected layers (FC). The output features from the FC layer are combined with trainable weights and biases and are reshaped into a 64 × 64 matrix.

The 64-dimensional local feature for each point is normalised by multiplying with the transformation matrix and is input into Classification Net and Segmentation Net. To determine whether a candidate bolt is a real rock bolt or not, Classification Net remaps the local features into 256 dimensions with shared MLP. It creates a 256-dimensional global feature for the entire candidate bolt with max pooling. We use max pooling to create the global feature vector because the input candidate rock bolt is a set of points without a specific order, and max is a symmetry function that can make the output result independent of the data feeding order. Finally, Classification Net uses a three-layer FC to calculate the bolt and not-bolt scores.

The Segmentation Net is similar to Classification layers except that it needs to predict a per point score and relies on both local and global features. Therefore, we concatenate the local features of each point with the global feature vector from Classification Net and creates a per-point vector in 320 dimensionalities. The vector is input into shared MLP to generate a 64-dimensional per-point feature. Another shared MLP finally processes the feature to calculate the per-point bolt and not-bolt scores.

To train RBNN, we split the candidate rock bolts generated in the rough estimation step into the training and testing dataset. During the training process, we use k-fold cross-validation. Specifically, we divide the training dataset into ten subsets. Each time we use nine subsets for training and one subset for validation. The testing dataset is used to evaluate the performance of RBNN after each training. In this study, 6093 candidate rock bolts are generated, and 80% of them are used for training while the remaining 20% are used for testing. For rock bolt classification, we label the candidate rock bolts as 1 if there is a real rock bolt and 0 otherwise. For rock bolt segmentation, each point in the candidate rock bolts is labelled according to its ground truth label in the original scan. We gradually increase the decay rate for batch normalisation from 0.6 to 0.99. We use adam optimiser with a learning rate 0.001, momentum 0.9, and batch size 48. The learning rate is decreased by half every 30 epochs. RBNN is implemented with TensorFlow [1] and takes about 2 hours to converge on an RTX 2080TI GPU.

The pseudocode for our proposed rock bolt detection algorithm `CFBolt` is given in Algorithm 1. For each point s_i in the input point cloud S, we first compute three eigenvalues using the PCA algorithm in the 10cm radius neighbouring ball centred at s_i (line 2). The POV vector (p_1, p_2) for s_i is calculated by normalizing the first two eigenvalues (line 3). Then a confidence score for each point is generated with LDA (line 4). In line 5, the point is labelled as 1 if the score is larger than threshold 0.9; otherwise, it is labelled as 0. In line 6, DBSCAN is performed on label 1 points to generate clusters. A 0.2m radius sphere centred at the central coordinate of each cluster is extracted as a candidate rock bolt (line 7). Finally, candidate rock bolts are fed into our proposed rock bolt detection neural network RBNN for classification and segmentation (line 8).

Algorithm 1 `CFBolt`

```
  Input: point cloud $S$
  Output: the rock bolt detection result $R$
1 for each point $s_i \in S$ do
2     $\lambda_1, \lambda_2, \lambda_3 \leftarrow \mathrm{PCA}(s_i)$
3     $p_1 \leftarrow \frac{\lambda_1}{\lambda_1+\lambda_2+\lambda_3}; p_2 \leftarrow \frac{\lambda_2}{\lambda_1+\lambda_2+\lambda_3}$
4     $d_i \leftarrow \mathrm{LDA}(p_1, p_2)$
5     $l_i \leftarrow 1$ if $d_i > 0.9$ else 0
6 $L \leftarrow \mathrm{DBSCAN}(\{s_i | s_i \in S \wedge l_i = 1\})$
7 $\mathcal{C} \leftarrow$ {extract a 0.2m radius sphere centred at each
    cluster in $L$}
8 $R \leftarrow \mathrm{RBNN}(C)$ for each candidate rock bolt $C \in \mathcal{C}$
9 return $R$
```

IV. EXPERIMENT RESULTS AND DISCUSSION

This section presents our experimental results. All experiments are performed under a Linux operating system on a machine with an Intel Xeon 3.4GHz CPU, NVIDIA RTX2080TI GPU, and 64GB RAM.

A. PERFORMANCE OF ROUGH ESTIMATION

We first report the efficiency and effectiveness of the rough estimation step. To demonstrate that using single-scale POV is a good choice for the rough estimation step, we report the performance of single-scale POV and multi-scale POV in Figure 8 and Table 2. The number after POV indicates the number of different scales used during computation. POV1 is the single-scale POV computed at a 10cm scale. POV5 and POV9 are multi-scale POV. POV5 is computed at 2cm to

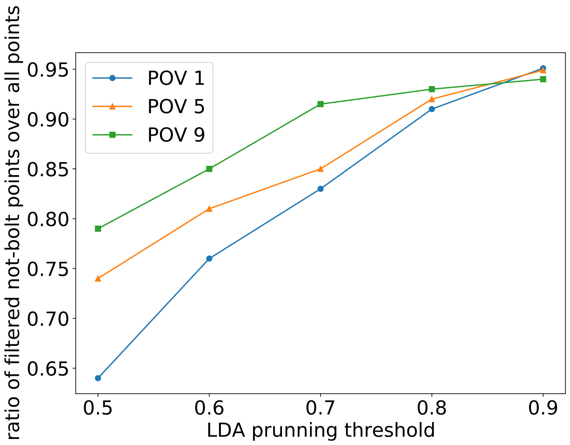


FIGURE 8. Filtering power of rough estimation with POV computed on different number of scales. POV1 is computed at 10cm scale. POV5 and POV9 are computed at 2cm to 10cm scale with the step by 2cm and 1cm, respectively.

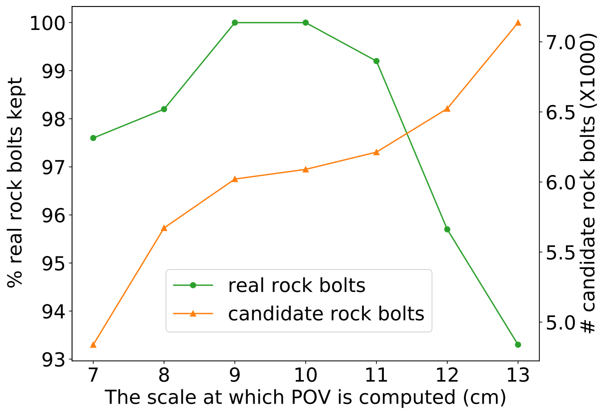


FIGURE 9. Performance of the rough estimation step when using single-scale POV at different scales. Green line indicates the percentage of real rock bolts kept after pruning. Red line indicates the number of candidate rock bolts generated in the rough estimation step.

TABLE 2. Performance of rough estimation with POV based on different scales, the pruning threshold is set to 0.9.

	running time	# candidate bolts	% real rock bolts kept
POV1	15.2s	72.5	100%
POV5	58.9s	73.0	100%
POV9	109.3s	74.3	100%

10cm scale with the step by 2cm, and POV9 is computed at 2cm to 10cm scale with the step by 1cm. With a low LDA pruning threshold, like 0.5, when the threshold increases, the difference reduces rapidly. When the threshold reaches 0.9, the filtering power of POV1 is even better than POV5 and POV9. The reason is that when the threshold is low, multi-scale POV enjoys advantages in capturing information on different scales. With the increase of the threshold, only the points having distinct bolt features will be kept. At this point, the difference between single-scale POV and multi-scale POV becomes insignificant. From Figure 8, we can conclude that the points with distinct rock bolt features can be effectively detected with single-scale POV. Furthermore, as shown in Table 2, it only takes 15.2s on average to perform the rough estimation with POV1 on a single scan. The average number of candidate rock bolts per scan with POV1 is the smallest, and no real rock bolt is filtered out.

In Figure 9, we report the results of using single-scale POV at different scales ranging from 7cm to 13cm increased by 1cm. When the scale is small or large, some real rock bolts are wrongly filtered out as the geometrical features of rock bolts are not captured accurately at these scales. Based on the experimental results, the best POV scales are 9cm and 10cm, at which all the real rock bolts are kept. The number of candidates rock bolts increases along with the scale. One of the reasons for this is that the geometrical features of the tunnel surface become evident at larger scales, making it more difficult to distinguish the rock bolts from the surface.

In summary, the rough estimation step with single-scale POV at 10cm scale satisfies the requirement that it filters as many not-bolt points as possible while all the real rock bolts are preserved.

B. ROCK BOLT DETECTION RESULT

We compare the performance of our algorithm `CFBolt` against three state-of-the-art deep learning algorithms (PointNet [27], RandLA-Net [15], Cylinder3D [41]) and the recent rock bolt detection algorithm (LDANN [34]). PointNet divides each tunnel scan into blocks and uniformly sample 4096 points in each block on the fly during training. For PointNet, we set the initial learning rate to 0:001 and the training epoch to 50 by default. RandLA-Net performs random samplings together with a local feature aggregation module on the input point cloud. For RandLA-Net, the initial learning rate is 0:01, and the training epoch is 100 by default. Cylinder3D, the newest work in general 3D object recognition, uses cylindrical partition to learn voxel-wise features and adopts asymmetrical 3D convolution networks. For RandLA-Net, we set the initial learning rate to 0:001 and the training epoch to 40 by default. LDANN is the newest algorithm dedicated to rock bolt detection. LDANN extracts per point features with POV and FPFH followed by a single-layer neural network to label each point.

The performance of all the above algorithms is assessed on the semantic segmentation (point-wise prediction accuracy). For algorithms dedicated to rock bolt detection, i.e., `CFBolt` and LDANN, we also assess the number of rock bolts correctly detected (classification).

The result of rock bolt semantic segmentation is given in Table 3. Following previous works [27], [34], we evaluate semantic segmentation with Intersection over Union (IoU) and precision. Let A be the predicted result and B be the

TABLE 3. Rock bolt semantic segmentation results.

	bolt (IoU %)	bolt (precision %)	not-bolt (IoU %)	not-bolt (precision %)
PointNet	52.11	58.60	98.11	98.87
RandLA-Net	73.45	77.23	99.25	99.48
Cylinder3D	65.32	71.42	98.83	99.21
LDANN	75.06	83.23	99.33	99.88
CFBolt	89.33	92.04	99.56	99.89

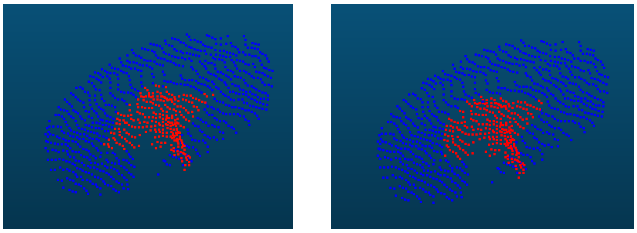


FIGURE 10. The semantic segmentation result of a candidate rock bolt. The red points in the left figure are the ground truth bolt points. The red points in the right figure are the bolt points labelled by CFBolt.

TABLE 4. Classification results.

	precision(%)	recall(%)	F1 score
LDANN	94.41	96.33	0.9536
CFBolt (POV1)	98.34	98.73	0.9853
CFBolt (POV5)	98.34	98.73	0.9853
CFBolt (POV9)	98.27	98.73	0.9850

ground truth. IoU and precision are expressed as $\frac{|A \cap B|}{|A \cup B|}$ and $\frac{|A \cap B|}{|A|}$, respectively. As shown in Table 3, our proposed rock bolt detection algorithm CFBolt outperforms others in terms of IoU and precision over bolt and not-bolt points. Specially, CFBolt improves 14% IoU and 8.8% precision against the newest rock bolt detection algorithm LDANN. It is unsurprising that existing deep learning algorithms, PointNet, RandLA-Net, and Cylinder3D, do not achieve the same performance as CFBolt because they are not tailored for tiny object detection, like rock bolts, and the intricate structure of rock bolt is not completely preserved during the downsampling process. Since 99.8% of the entire scan are not-bolt points, all algorithms achieve more than 98% IoU and precision regarding not-bolt points, and the difference is relatively small. Nevertheless, CFBolt still achieves the best result with 99.56% IoU and 99.89% precision for not-bolt points.

We visualize the semantic segmentation result of our proposed rock bolt detection algorithm CFBolt in Figure 10. Compared with the ground truth, our semantic segmentation result only has a small difference at the corner of the rock bolt. Compared with the rock bolt semantic segmentation by LDANN in Figure 2, which only detects the circular region on the bearing plate, the improvement by our algorithm is significant.

To evaluate the result of rock bolt classification, we report the precision, recall, and F1 score for LDANN and CFBolt in Table 4. F1 score is the $\frac{2 \times precision \times recall}{precision + recall}$, which conveys the balance between the precision and the recall. We report the results of CFBolt using POV1, POV5, and POV9 separately to show the influence of POV on the rock classification. As shown in Table 4, in terms of the recall rate, there is no difference between multi-scale POV (POV5 and POV9) and single-scale POV (POV1). This is because all the real rock bolts are kept after the rough estimation step, and the number of real rock bolts that are correctly detected by our neural network is the same. The precision of POV9 is lower than POV1 as more candidate bolts are generated with POV9 (see Table 2), and some are misclassified by the neural network. Our proposed rock bolt detection algorithm CFBolt (POV1) outperforms LDANN in terms of precision and recall by 3.9% and 2.4%, respectively. The F1 score is improved by 3.3% over LDANN from 0.9536 to 0.9853. Overall, the experiment results demonstrate our two-step coarse-to-fine strategy's superiority and the effectiveness of using deep neural networks for rock bolt detection.

C. RUNNING TIME

We finally report the average running time for CFBolt on each scan in Table 5. To have a better overall view of our algorithm, we divide CFBolt into four parts and report time cost for each of them. As shown in the table, using single-scale POV (CFBolt (POV1)) significantly reduces the time cost for the LDA part. Furthermore, as more candidate rock bolts are generated by the multi-scale POV (POV5 and POV9), it takes longer to process the data in all following steps. The total running time for processing one scan with single-scale POV is 56s, while it takes more than 100s using multi-scale POV. The time cost of generating candidate rock bolts is relatively high because we need to conduct a range query in the raw scan for each candidate rock bolt.

TABLE 5. Running time.

Step	Avg. running time (s)		
	CFBolt (POV1)	CFBolt (POV5)	CFBolt (POV9)
LDA (line 1-5)	18.83	63.30	114.57
DBSCAN (line 6)	0.71	0.72	0.74
Generating candidate rock bolts (line 7)	27.28	27.47	27.96
RBNN (line 8)	9.64	9.71	9.88
Total	56.46	101.02	152.66

V. CONCLUSION

This paper presents an automatic algorithm for detecting supporting rock bolts from LiDAR scan data. The proposed algorithm follows a coarse-to-fine strategy to process the LiDAR data. It first adopts a simple but effective local feature to filter out as many not-bolt points as possible. Then, inspired by the success of deep-learning-based 3D object detection algorithms, a deep neural network is designed and trained to further process the pruned point cloud. Experiments on real tunnel scan data show the superiority of our algorithm in terms of both effectiveness and efficiency. The rock bolt detection results can assist in predictive maintenance and provide significant advances in the reduction of labour through automation. Further work will focus on fine-tuning our proposed algorithm for mobile edge computing to achieve in-situ real-time data processing.

REFERENCES

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, and C. Citro, "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," 2016, *arXiv:1603.04467*. [Online]. Available: https://arxiv.org/abs/1603.04467

[2] S. Balakrishnama and A. Ganapathiraju, "Linear discriminant analysis-a brief tutorial," *Inst. Signal Inf. Process.*, vol. 18, pp. 1–8, Mar. 1998.

[3] D. Benton, J. Seymour, M. S. Boltz, M. Raffaldi, and S. Finely, "Photogrammetry in underground mining ground control—Lucky Friday mine case study," in *Proc. 8th Int. Conf. Deep High Stress Mining*, 2017, pp. 587–598.

[4] R. Blomley, B. Jutzi, and M. Weinmann, "Classification of airborne laser scanning data using geometric multi-scale features and different neighbourhood types," *ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 3, pp. 169–176, Jun. 2016.

[5] N. Brodu and D. Lague, "3D terrestrial lidar data classification of complex natural scenes using a multi-scale dimensionality criterion: Applications in geomorphology," *ISPRS J. Photogram. Remote Sens.*, vol. 68, pp. 121–134, Mar. 2012.

[6] E. Brown and E. Hoek, *Underground Excavations Rock*. Boca Raton, FL, USA: CRC Press, 1980.

[7] B. J. Buys, P. S. Heyns, and P. Loveday, "Rock bolt condition monitoring using ultrasonic guided waves," *J. Southern Afr. Inst. Mining Metall.*, vol. 109, no. 2, pp. 95–105, 2009.

[8] T. J. B. Dewez, S. Yart, Y. Thuon, P. Pannet, and E. Plat, "Towards cavity-collapse hazard maps with Zeb-Revo handheld laser scanner point clouds," *Photogramm. Rec.*, vol. 32, no. 160, pp. 354–376, Dec. 2017.

[9] A. F. C. Errington, B. L. F. Daku, and A. Prugger, "Closure monitoring in potash mines using LiDAR," in *Proc. 36th Annu. Conf. Ind. Electron. Soc.*, Nov. 2010, pp. 2823–2827.

[10] B. V. Farahani, F. Barros, P. J. Sousa, P. P. Cacciari, P. J. Tavares, M. M. Futai, and P. Moreira, "A coupled 3D laser scanning and digital image correlation system for geometry acquisition and deformation monitoring of a railway tunnel," *Tunnelling Underground Space Technol.*, vol. 91, Sep. 2019, Art. no. 102995.

[11] J. R. Gage, D. Fratta, A. L. Turner, M. M. MacLaughlin, and H. F. Wang, "Validation and implementation of a new method for monitoring *in situ* strain and temperature in rock masses using fiber-optically instrumented rock strain and temperature strips," *Int. J. Rock Mech. Mining Sci.*, vol. 61, pp. 244–255, Jul. 2013.

[12] J. Gallwey, M. Eyre, and J. Coggan, "A machine learning approach for the detection of supporting rock bolts from laser scan data in an underground mine," *Tunnelling Underground Space Technol.*, vol. 107, Jan. 2021, Art. no. 103656.

[13] V. Gikas, "Three-dimensional laser scanning for geometry documentation and construction management of highway tunnels during excavation," *Sensors*, vol. 12, no. 8, pp. 11249–11270, Aug. 2012.

[14] J.-Y. Han, J. Guo, and Y.-S. Jiang, "Monitoring tunnel deformations by means of multi-epoch dispersed 3D LiDAR point clouds: An improved approach," *Tunnelling Underground Space Technol.*, vol. 38, pp. 385–389, Sep. 2013.

[15] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham, "RandLA-Net: Efficient semantic segmentation of large-scale point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11108–11117.

[16] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," 2015, *arXiv:1506.02025*. [Online]. Available: http://arxiv.org/abs/1506.02025

[17] C. C. Li, "Principles of rockbolting design," *J. Rock Mech. Geotechnical Eng.*, vol. 9, no. 3, pp. 396–414, Jun. 2017.

[18] D. Li, Q. Xie, X. Gong, Z. Yu, J. Xu, Y. Sun, and J. Wang, "Automatic defect detection of metro tunnel surfaces using a vision-based inspection system," *Adv. Eng. Informat.*, vol. 47, Jan. 2021, Art. no. 101206.

[19] J. Li, B. M. Chen, and G. H. Lee, "So-Net: Self-organizing network for point cloud analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9397–9406.

[20] L. Li, P. C. Hagan, S. Saydam, B. Hebblewhite, and Y. Li, "Parametric study of rockbolt shear behaviour by double shear test," *Rock Mech. Rock Eng.*, vol. 49, no. 12, pp. 4787–4797, Dec. 2016.

[21] L. Li, P. C. Hagan, S. Saydam, B. Hebblewhite, and C. Zhang, "A laboratory study of shear behaviour of rockbolts under dynamic loading based on the drop test using a double shear system," *Rock Mech. Rock Eng.*, vol. 52, no. 9, pp. 3413–3429, Sep. 2019.

[22] J. Martínez-Sánchez, I. Puente, H. GonzálezJorge, B. Riveiro, and P. Arias, "Automatic thickness and volume estimation of sprayed concrete on anchored retaining walls from terrestrial lidar data," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 5, pp. 521–526, Jun. 2016.

[23] P. Morissette, J. Hadjigeorgiou, A. Punkkinen, and D. Chinnasane, "The influence of change in mining and ground support practice on the frequency and severity of rockbursts," in *Proc. 7th Int. Conf. Deep High Stress Mining*, 2014, pp. 165–177.

[24] F. Öberg, "Method and system for monitoring and documenting installation of rock reinforcement bolt," U.S. Patent 8 606 542, Dec. 10, 2013.

[25] S. S. Peng and D. H. Y. Tang, "Roof bolting in underground mining: A state-of-the-art review," *Int. J. Mining Eng.*, vol. 2, no. 1, pp. 1–42, Mar. 1984.

[26] Y. Potvin and J. Hadjigeorgiou, "Selection of ground support for mining drives based on the q-system," in *Proc. 8th Int. Symp. Ground Support Mining Underground Construction*, 2016, pp. 1–16.

[27] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, Jul. 2017, pp. 652–660.

[28] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," 2017, *arXiv:1706.02413*. [Online]. Available: http://arxiv.org/abs/1706.02413

[29] M. M. Rahman, Y. Tan, J. Xue, and K. Lu, "Notice of violation of IEEE publication principles: Recent advances in 3D object detection in the era of deep neural networks: A survey," *IEEE Trans. Image Process.*, vol. 29, pp. 2947–2962, 2020.

[30] R. B. Rusu, "Semantic 3D object maps for everyday manipulation in human living environments," *KI-Künstliche Intell.*, vol. 24, no. 4, pp. 345–348, 2010.

[31] S. P. Signer, D. Cox, and J. Johnston, "A method for the selection of rock support based on bolt loading measurements," in *Proc. 16th Int. Conf. Ground Control Mining*, 1997, pp. 183–190.

[32] S. Kumar Singh, B. Pratap Banerjee, and S. Raval, "Three dimensional unique identifier based automated georeferencing and coregistration of point clouds in underground environment," 2021, *arXiv:2102.10731*. [Online]. Available: http://arxiv.org/abs/2102.10731

[33] S. K. Singh, S. Raval, and B. Banerjee, "A robust approach to identify roof bolts in 3D point cloud data captured from a mobile laser scanner," *Int. J. Mining Sci. Technol.*, vol. 31, no. 2, pp. 303–312, Mar. 2021.

[34] S. K. Singh, S. Raval, and B. Banerjee, "Roof bolt identification in underground coal mines from 3D point cloud data using local point descriptors and artificial neural network," *Int. J. Remote Sens.*, vol. 42, no. 1, pp. 367–377, Jan. 2021.

[35] S. K. Singh, S. Raval, and B. P. Banerjee, "Automated structural discontinuity mapping in a rock face occluded by vegetation using mobile laser scanning," *Eng. Geol.*, vol. 285, May 2021, Art. no. 106040.

[36] M. Soilán, A. Sánchez-Rodríguez, and P. del Río-Barral, "Review of laser scanning technologies and their applications for road and railway infrastructure monitoring," *Infrastructures*, vol. 4, no. 4, p. 58, Sep. 2019.

[37] K. Tan, X. Cheng, Q. Ju, and S. Wu, "Correction of mobile TLS intensity data for water leakage spots detection in metro tunnels," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 11, pp. 1711–1715, Nov. 2016.

[38] T. Xu, L. Xu, X. Li, and J. Yao, "Detection of water leakage in underground tunnels using corrected intensity data and 3D point cloud of terrestrial laser scanning," *IEEE Access*, vol. 6, pp. 32471–32480, 2018.

[39] Y. Xu, D. Li, Q. Xie, Q. Wu, and J. Wang, "Automatic defect detection and segmentation of tunnel surface using modified mask R-CNN," *Measurement*, vol. 178, Jun. 2021, Art. no. 109316.

[40] C. Yi, D. Lu, Q. Xie, J. Xu, and J. Wang, "Tunnel deformation inspection via global spatial axis extraction from 3D raw point cloud," *Sensors*, vol. 20, no. 23, p. 6815, Nov. 2020.

[41] H. Zhou, X. Zhu, X. Song, Y. Ma, Z. Wang, H. Li, and D. Lin, "Cylinder3D: An effective 3D framework for driving-scene LiDAR semantic segmentation," 2020, *arXiv:2008.01550*. [Online]. Available: http://arxiv.org/abs/2008.01550

SARP SAYDAM received the bachelor's degree (Hons.) in mining engineering from the School of Minerals and Energy Resources Engineering, UNSW Sydney. He is currently working as a Digital Development Engineer with DSI Underground, Sydney, Australia. He has three years of experience as an Underground LiDAR Scan Technician.

BOGE LIU received the B.E. degree from the School of Software, Tsinghua University, China, in 2016, and the Ph.D. degree in computer science and engineering from the University of New South Wales, Sydney, Australia, in 2020. He is currently a Postdoctoral Fellow with the School of Minerals and Energy Resources Engineering, University of New South Wales, Sydney. His research interests include big data analytics, database, and point cloud data processing.

BINGHAO LI (Member, IEEE) received the B.E. degree in electrical and mechanical engineering from Beijing Jiaotong University, China, in 1994, the M.E. degree in civil engineering from Tsinghua University, China, in 2001, and the Ph.D. degree from the University of New South Wales, Sydney, Australia, in 2006. He is currently a Senior Lecturer with the MIoT and IPIN Laboratory, School of Minerals and Energy Resources Engineering, University of New South Wales, Sydney. His research interests include indoor positioning, pedestrian navigation, satellite positioning and navigation, and mine Internet of Things.

WENJIE ZHANG is currently a Professor and the Deputy Head of the School (Research) with the School of Computer Science and Engineering, University of New South Wales, Sydney, Australia. Her research interests include spatial-temporal data analysis, uncertain data analysis, and graph data processing. Since 2008, she has been published more than 100 papers in top venues, such as IEEE Transactions on Knowledge and Data Engineering (TKDE), *ACM TODS*, *VLDB Journal*, SIGMOD, VLDB, and ICDE. In 2011, she received the Australian Research Council Discovery Early Career Researcher Award. In 2019, she received the Prestigious Chris Wallace Award for her significant contributions to large-scale graph data processing. Her research has been well supported by seven Australian Research Council grants and several other UNSW and industry funding, totally over $2.8 million, since 2012. She is an Associate Editor for IEEE Transactions on Knowledge and Data Engineering.

SARVESH KUMAR SINGH is currently pursuing the Ph.D. degree with the School of Minerals and Energy Resources Engineering (MERE), University of New South Wales, Sydney, Australia. He is working on the accurate 3D mapping of an underground environment, including mines and tunnels. The project involves laser scanning, simultaneous localisation and mapping (SLAM), and machine learning.

SIMIT RAVAL is currently the Director of undergraduate studies in mining engineering and the Co-Director of the Laboratory for Imaging of the Mine Environment (LIME), UNSW, Sydney. He is specialised in the integration of sensing technologies to drive applied innovation in mining, environmental, and civil engineering sectors. He leads a group of researchers focused on utilising data from sensors mounted on various platforms, from satellite through to UAVs, to visualise, identify, and monitor operational environments. He has received more than $2.8 million as competitive research grants, including four nationally competitive Australian Coal Association Research Program (ACARP) grants the project leader. He has supervised Ph.D. research projects involving drone-based smart sensing (hyperspectral and LiDAR), underground mobile laser scanning, image-based automated material characterisation, mine rehabilitation/closure, climate change, and asteroid mining.