ITC 4/53
Information Technology and Control
Vol. 53 / No. 4/ 2024
pp. 983-996
DOI 10.5755/j01.itc.53.4.37475

Verification of 3D Electrical Equipment Model Based on Cross-source Point Cloud Registration Using Deep Neural Network

Received 2024/05/30 | Accepted after revision 2024/08/01

HOW TO CITE: Yu, H., He, Z., Peng, L., Zhou, A. (2024). Verification of 3D Electrical Equipment Model Based on Cross-source Point Cloud Registration Using Deep Neural. *Information Technology and Control*, 53(4), 983-996. https://doi.org/10.5755/j01.itc.53.4.37475

Verification of 3D Electrical Equipment Model Based on Cross-source Point Cloud Registration Using Deep Neural Network

Hai Yu
Information and Communication Research Institute of China Electric Power Research Institute Co., Ltd, Nanjing, Jiangsu, 210000, China;
School of Automation, Southeast University, Nanjing, Jiangsu, 210096, China

Zhimin He, Lin Peng, Aihua Zhou
Information and Communication Research Institute of China Electric Power Research Institute Co., Ltd, Nanjing, Jiangsu, 210000, China

Corresponding author: 230209100@seu.edu.cn

With the popularization of digital twin techniques in power substations, assessment and verification of electrical equipment 3D models in digital twins according to as-built LiDAR point clouds become essential for the quality assurance of the designed substation models. However, computing the shape and texture differences between a 3D model and its corresponding point cloud is challenging due to the difficulty in aligning cross-source equipment point clouds with local geometric shape variations. In this paper, we propose a 3D model verification method based on overlap-aware cross-source point cloud registration. The key of the method is an overlap attention-based point cloud registration network with grouped KPConv, attention mechanism, and overlap-weighted circle loss. It improves the registration accuracy against local geometric shape variations between 3D models and LiDAR point clouds. In addition, due to the lack of real-world point cloud samples of electrical equipment, a novel point cloud augmentation method is employed for generating synthetic point clouds for improving the sim-to-real generalization capability of the network. Based on the pose alignment of the 3D model and the corresponding point cloud, a facet-level computing method is proposed for model differentiation and colorization. Experimental results using real-world point clouds of power substation equipment validate the performance of the proposed method.

KEYWORDS: Point cloud registration; Deep learning; Attention; 3D model verification; Electrical equipment.

1. Introduction

With the emergence of the Digital Twin (DT) technology in the power grid, the quality assessment of electrical equipment 3D models in a digital twin of power substation is increasingly important. In particular, the reality gap between the electrical equipment DT model and its corresponding as-built point cloud requires 3D model verification before the DT model can be transferred from the design department to the operation department of a power substation.

Currently, the verification of the designed substation equipment model mainly relies on manual checking according to regulations and specifications. This results in low efficiency in the verification of digital twins. Using the as-built point cloud obtained from LiDAR scans [22], the apparent difference between real-world equipment and its corresponding digital twin model can be computed. In this paper, we focus on automated apparent verification of electrical equipment models in terms of reality gaps in shape and texture, which are two major concerns in model verification tasks.

A key issue is that, as the real-world point clouds of electrical equipment are extracted from the LiDAR scan in a substation scene, the pose of equipment relative to its DT model is unknown. Therefore, the key prerequisite for the automatic verification of equipment models is to accurately align the point cloud of equipment to its model. This allows for a comparison between the two in the same scale and coordinate system. Although there is wide literature about point cloud registration methods, registering point clouds for electrical equipment poses significant challenges for the following reasons.

First, aligning a DT model with the corresponding point cloud requires cross-source registration [7, 10, 28]. However, the point distributions of the pairwise discretized model and LiDAR point cloud are uneven. Traditional FPFH [34], SIFT [14], DCA [4] and other feature-based registration methods are limited in accuracy because of the uneven point cloud density. In contrast, the deep learning-based registration methods outperform traditional hand-engineered feature-based methods in accuracy. However, the major challenge in deep learning-based point cloud registration methods for electrical equipment is the lack of point cloud datasets for training registration networks, which hinders their generalization ability.

Second, due to the complex geometric structures of electrical equipment, there are inevitable local geometric shape deviations between DT models and LiDAR point clouds. While the source and target point clouds are assumed identical for most existing point cloud registration methods, they are partially different in our case. For traditional point cloud registration methods such as ICP [8] and most deep-learning-based methods [16, 1, 2], such local shape deformation will confuse the registration process, ultimately resulting in a decrease in registration accuracy. Therefore, awareness of the overlap region is essential for electrical equipment point cloud registration.

Another problem is shape and texture assessment for the aligned DT model and point cloud. The reality gaps in shape and texture are two major concerns in model verification because the shape and texture variances between real-world equipment and its DT model are mostly common. As a 3D model can be discretized by sampling, a straightforward approach is to directly differentiate the point cloud generated from the 3D model discretization and the LiDAR point cloud. However, direct point cloud differentiation will lead to shape-irrelevant errors due to the uneven point density distribution of the two sources of point clouds.

In this paper, a DT electric equipment model verification method is proposed based on overlap-aware cross-source point cloud registration. The key of the method is a deep point cloud registration network with grouped KPConv, attention mechanism, and overlap-weighted circle loss for aligning cross-source point clouds of electrical equipment. The network improves the registration robustness against local geometric shape variations between the DT model and its LiDAR point cloud. Using the registration result, a model differentiation and colorization method based on facet-point correspondence is also proposed.

The rest of this paper is organized as follows. In Section 3, the overall method is briefly described. In Section 4, the proposed GKANet network for point cloud registration is described in detail, and in Section 5 the model verification method is described. In Section 6, the experiments are described with result analysis, followed by a conclusion of the paper.

2. Related Works

The verification of 3D models of electrical equipment is crucial because of the deviation between the electrical equipment 3D models and their as-built point cloud. This is an important quality assurance process of the designed substation models. In addition, 3D model verification of electrical equipment also plays a significant role in equipment status monitoring and maintenance, as some electrical equipment such as power pylons may have the risk of shape deformation during long-term operation.

Compared with time-consuming and laborious manual verification, automated model verification provides a more appealing approach. A similar task is point cloud-based deformation analysis, which aims to determine geometric changes of objects related to a reference model. In [31], a novel method for measuring 3D deformation based on image-guided point cloud registration is proposed. However, [31] only focuses on in-plane 3D displacement while we emphasize estimating the 3D rotational and 3D translational displacement.

Although point cloud-based deformation analysis has been widely applied to structural objects [33, 25, 19] and industrial objects, its application in smart grids is relatively rare. In [16], the measurement and evaluation of the shape deformation of electrical equipment is achieved by a local invariant feature-based method. However, the method in [16] does not consider the partial overlap problem and the registration method remains the conventional SAC-IA algorithm for coarse registration, followed by ICP [8] for fine registration.

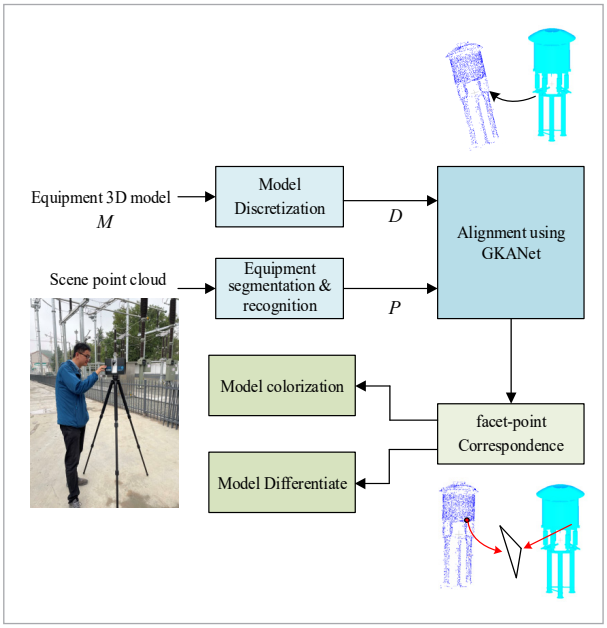
At present, non-deep-learning-based algorithms of point cloud registration include point-based methods and feature-based methods. The point-based registration methods, such as ICP [8], 4PCS [29], Super4PCS [12], CICP [9], and Teaser++ [27], are generally more computationally costly. In contrast, the feature-based registration methods are more efficient in computation. They extract FPFH [22], SIFT [14], DCA [4], or other semantic features [19] for matching the correspondence points and registering the point clouds.

Meanwhile, as deep learning methods have shown advantages in point cloud analysis [18, 3], they are also promising in point cloud registration. Deep learning-based methods [24, 5, 1, 6] exhibit tremendous potential in registering point clouds that contain repetitive and symmetrical features, because of their capability to learn more robust feature representations. PCRNet [15] uses PointNet for extracting global features. Compared with PointNetLK [1], PCRNet has better generalization capability but is less robust to noise. D3Feat [2] is developed based on the point cloud feature extraction network KPConv to compute local features. PREDATOR [6], also based on D3Feat, combines the Transformer attention mechanism and designs an overlapping attention module. Thanks to the use of cross-attention and self-attention mechanisms, PREDATOR can simultaneously obtain both local and global information on point clouds, inferring which points may be located in overlapping regions.

3. Method Overview

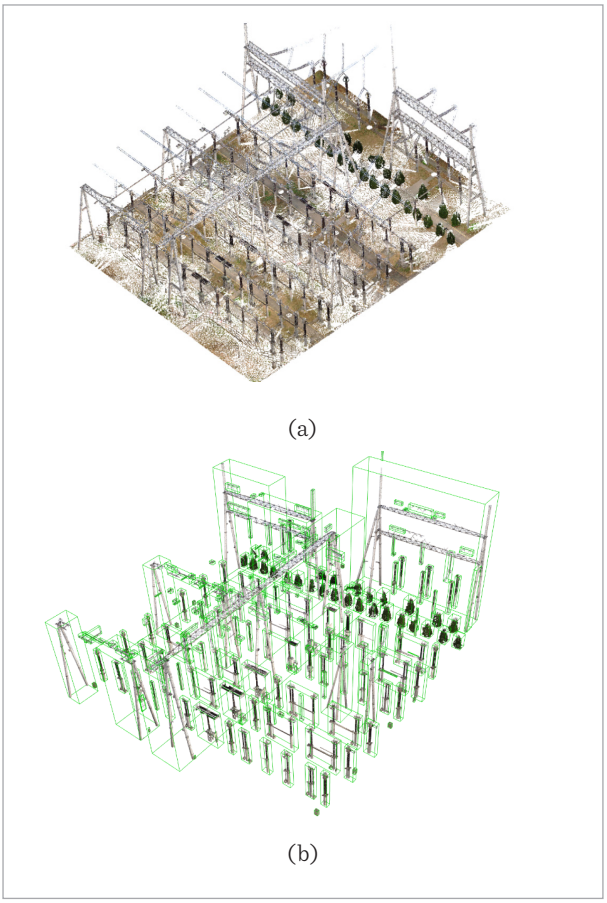
The overview of the model verification method is depicted in Figure 1. Using the point cloud generated by the 3D LiDAR sensory system, electric equipment is segmented from the substation point cloud and recognized as one of the objects in the model library.

Figure 1
Overview of the method



This is implemented by ground plane removal, power line removal, octree-based connected-component labeling for segmentation, and 3D shape recognition. In particular, the shape recognition is achieved by extracting the ESF descriptor [17] of an equipment point cloud and comparing it with that of all models in the model library. Figure 2(a) shows an example of a power substation and Figure 2(b) is the result of electric equipment segmentation.

Figure 2
Extracting electrical equipment point cloud from a substation scene



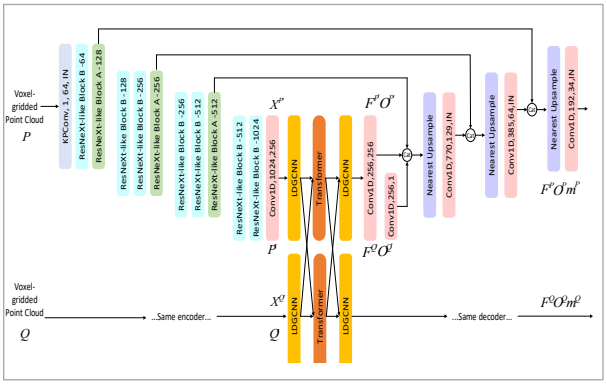
The key of the method is a point cloud registration network, GKANet, that aligns the pairwise model and point cloud of specific electric equipment. Based on the registration result, the model differentiation and colorization are achieved by establishing correspondence between a triangular facet in the model and the corresponding points in the point cloud.

4. GKANet for Point Cloud Registration

The overall structure of the registration network is divided into three main parts: the Encoder, Overlap Attention Module, and Decoder. The role of the Encoder is to effectively encode and compress the two point clouds P and Q for registration, resulting in a set of superpoints P' and Q' that represent the key features of the original data, as well as their respective feature descriptors $X^{P'}$ and $X^{Q'}$. The Overlap Detection Module uses LDGCNN to extract features and project them onto the overlap scores and cross-overlap scores, computed on superpoints.

The Decoder is responsible for translating the conditional features and overlap scores into feature descriptors, overlap scores, and matchability scores for each point. The overall structure of the GKANet network is illustrated in Figure 3.

Figure 3
The overall structure of the proposed GKANet



4.1. Encoder with Grouped KPConv

The encoder begins with a voxel grid filter to downsample the raw point cloud. This step produces a source point cloud P and a target point cloud Q with reasonable point density. This effectively reduces the complexity and computational load of subsequent processing. Then the encoder uses a set of grouped ResNet blocks and strided convolutions to extract features from P and Q, and perform downsampling, aggregating the raw points into superpoint sets P' and Q'. In this way, each superpoint can represent a patch within the point cloud.

Compared with [6], the structure of grouped convolutions [26] with Kernel Point Convolution [21] is introduced. The structure is characterized by the splitting-transforming-aggregating mechanism. Figure 4 shows the structure of a ResNeXt block with KPConv, characterized by the splitting-transforming-aggregating mechanism. For an input with n channels, 32 convolution path groups are split to sequentially perform 1X1, 3X3, and 1X1 convolutions, then aggregating to output n/2 channels. This grouped convolution structure effectively enhances the network's performance while reducing the computational load. KPConv [21] is used to convolve directly in three-dimensional space, thus better capturing the spatial structure and features of the point clouds. GAM (Global Attention Mechanism) [11] is employed to enable the encoder to more effectively discern the essential characteristics within the point cloud data.

Using the encoder, point clouds P and Q are aggregated into superpoint sets P' and Q' after passing through the encoder composed of stacked ResNeXt blocks. $X^{P'}$ and $X^{Q'}$ represent the corresponding features, respectively.

4.2. Overlap Attention Module

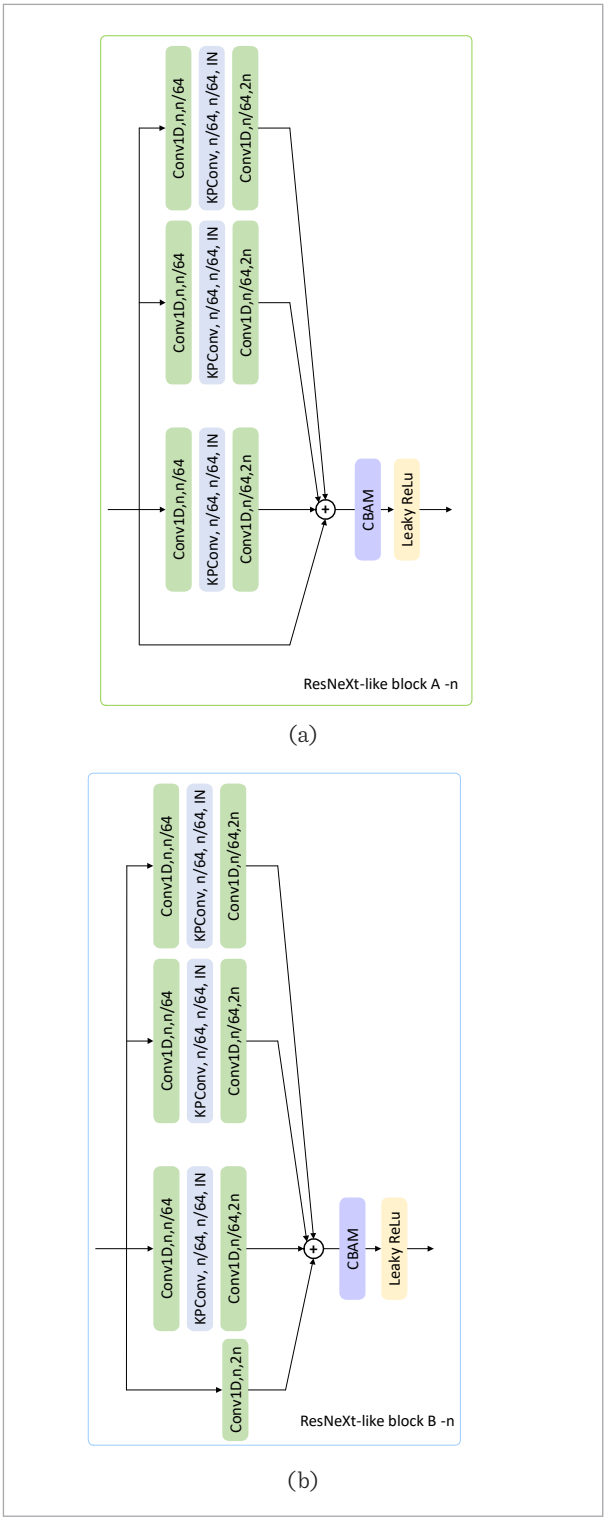
Before establishing connections between the source superpoints and the target superpoints, the Linked Dynamic Graph CNN (LDGCNN) [32] is employed to extract features. Compared with DGCNN [23], LDGCNN links the features of different levels of the dynamic graph for capturing edge features. Linking hierarchical structure features can avoid gradient vanishing in deep neural networks. It also removes the transformation network in DGCNN and uses MLP instead for extracting transform invariant features. Therefore, LDGCNN reduces the complexity of the network and improves the computational speed, compared with DGCNN.

For a point p_i, a local feature f_i is extracted on the local graph centralized on p_i with the edges e_i between the central point and its K neighbours.

$$f_i = f_e(G(p_i, e_i)) = \max\{h(p_i, e_{i_1}), h(p_i, e_{i_2}), \ldots, h(p_i, e_{i_K})\} \quad (1)$$

LDGCN extracts the hidden feature vector (p_i, e_{i_j}) using MLP with parameters $w_{c'c}$, $w_{c'(c+C)}$ and $b_{c'}$

Figure 4
ResNeXt block structure with KPConv



$$h(p_i, e_{i_1}) = (h_1(p_i, e_{i_j}), h_2(p_i, e_{i_j}), ..., h_{C'}(p_i, e_{i_j})) \quad (2)$$

$$h_{c'}(p_i, e_{i_j}) = \sum_{c=1}^{C} w_{c'c} p_{ic} + w_{c'(c+C)}(p_{i_j c} - p_{ic}) + b_{c'} \quad (3)$$

The overlap detection module determines whether the superpoints from two separate point clouds overlap using pairwise comparisons. If a superpoint i in P' matches a superpoint in Q', we may assume that superpoint i is likely to be situated within the overlapping region.

We leverage the self-attention mechanism of Transformer to enable each superpoint to exploit the information from all other superpoints, thereby better determining whether it is in the overlap area. In particular, the query vector Q, key vector K and value vector V can be defined as follows:

$$Q = W^Q F_X, K = W^K F_X, V = W^V F_X, \quad (4)$$

where W^Q, W^K and W^V are learnable weight matrices, F_X is the feature of the point cloud X. Then, we obtain a co-contextual feature z_i using the attention mechanism of the Transformer model. This new feature vector contains shared information about the superpoint sets P' and Q' and can reflect whether the superpoint is located in the overlap region. The new feature vector z_i is then explicitly updated for local context information through an LDGCNN with a similar structure, resulting in the final latent feature encoding. The latent feature encodings of all superpoints in P' are denoted as $F^{P'}$. Similarly, we use cross-attention to obtain new feature vectors for each superpoint in Q', and the final latent feature encoding is denoted as $F^{Q'}$. Furthermore, these latent feature encodings are linearly projected onto the overlap score $O^{P'}$ and $O^{Q'}$, and cross-overlap score $\tilde{O}^{P'}$ and $\tilde{O}^{Q'}$. The overlap score o_i of the superpoint i is a value between 0 and 1, which indicates the probability that the superpoint is located in the overlap region. The overlap score is computed according to Equation (5), in which w_o^T represents the weight vector. Therefore, we can determine the overlap area of point clouds P and Q based on the overlap score.

$$o_i = sigmoid(w_o^T z_i) \quad (5)$$

4.3. Decoder

For P', the inputs for the decoder are the feature encoding of the superpoints $F^{P'}$ and the overlap score $O^{P'}$. The outputs are the pointwise feature descriptors F^P, the point-wise overlap scores O^P, and the matching scores m^P. Similarly, for Q', the inputs for the decoder are $F^{Q'}$ and $O^{Q'}$, and the outputs are the point-wise feature descriptors F^Q, the overlap scores O^Q, and the matching scores m^Q. For the detailed structure of the Decoder, please refer to [6].

4.4. Loss Function

We improve the Circle Loss in [6] by introducing a new Overlap-weighted Circle Loss, which assigns different weights to positive samples according to their overlap scores. To calculate the overlap scores, we define a set of anchor patches A, where for each point in P, there is at least one positive sample in Q. According to [13], if two point patches have at least 10% overlap, we consider them as positive samples; if they have no overlap, we consider them as negative samples. All other pairings are ignored. This loss function enables the model to focus on those matches with high overlap, thereby improving the model's performance.

For each anchor patch $G_{P_i} \in A$, its positive sample set in Q is denoted as ϵ_{i_p}, and the set of negative samples is denoted as ϵ_{i_n}, respectively. Then, the Overlap-weighted Circle Loss $L_{P_{oc}}$ on P is defined as follows:

$$L_{P_{oc}} = \frac{1}{|A|} \sum_{G_{P_i} \in A} log[1 + \sum_{G_{Qj} \in \epsilon_{i_p}} e^{\lambda_{j_i} \beta_{i,j_p} (d_{j_i} - \Delta_p)} \sum_{G_{Qk} \in \epsilon_{i_n}} e^{\beta_{i,k_n} (\Delta_n - d_{k_i})}] \quad (6)$$

In Equation (6), $d_{j_i} = \| \hat{h}_{P_i} - \hat{h}_{Q_j} \|^2$ measures the distance in the feature space, o_{j_i} is the overlap ratio between G_{P_i} and G_{Q_j}, and $\lambda_{j_i} = (o_{j_i})^{1/2}$. The weights for positive samples and negative samples are calculated individually for each sample, according to $\beta_{i,j_p} = \gamma(d_{j_i} - \Delta_p)$ and $\beta_{i,k_n} = \gamma(\Delta_n - d_{k_i})$. The margin hyperparameters are set as $\Delta_p = 0.1$ and $\Delta_n = 1.4$.

Overlap-weighted Circle Loss reweights the loss values on ϵ_{i_p} according to the overlap ratio, thus giving higher importance to patches with higher overlap.

The same method is applied to the loss on Q, which is denoted as $L_{Q_{oc}}$. The total loss function is:

$$L_{oc} = (L_{P_{oc}} + L_{Q_{oc}}) / 2 \quad (7)$$

Apart from the Overlap-weighted Circle Loss L_{oc}, two other loss functions are also employed, including the Overlap Loss L_o [6], and the Matchability Loss L_m [6]. The total loss is formed by a weighted sum of these three losses:

$$L = L_{oc} + \lambda_o L_o + \lambda_m L_m \quad (8)$$

Although we found that the Circle Loss [21] and the Overlap Loss are helpful for network convergence in the early stages of training, the Matchability Loss becomes more important in the later stages as it directly influences the final matching performance. Therefore, we gradually increase the value of λ_m during the training to augment the weight of the Matchability Loss.

5. Model Verification Based on Cross-source Point Cloud Registration

5.1. Model Differentiate

Model differentiation is performed to highlight the differences between the model and the point cloud of equipment with distinct colors. The process of model differencing is shown in Algorithm 1.

Denote M as the equipment models and P as the corresponding point cloud. First, the equipment models are discretized using grid sampling based on Poisson-disk distribution, resulting in the target point cloud D. We then perform point cloud registration between P and D to obtain the alignment transform. As shown in Line 3, we then transform P to the coordinate of D to align the two, resulting in a transformed point cloud P_t. In Line 4 to 5, a triangular facet $Plane_j$ is selected from M, in which N points $\{n_i\}, i = 1, ..., N$ are randomly sampled. In line 7 to 11, for each point n_i, a neighborhood search with radius r is conducted in P_t. In Line 13 to 15, if the number of sampled points with neighboring points is lower than a threshold λN, $Plane_j$ is considered as a redundant facet and highlighted in red. In Line 17 to 22, for each point in P_t, a neighborhood radius search is performed in D. If no neighboring points are found, the point is highlighted in red as a missing part in the model.

Algorithm 1: Model and Point Cloud Differentiate

Input: M: Equipment model to be verified; P: Corresponding point cloud; r: search radius;

Output: Highlight the difference between the model and the point cloud

1: $D = PoissonDisk(M)$
2: $T = Align(D, P)$
3: $P_t = T * P$
4: For each $Plane_j$ in M do
5: Randomly sample N *points*
6: The number of sample points with neighbors in P_t: $Count = 0$
7: For each sample point n_i do
8: Neighborhood radius search in P_t
9: IF *Neighborhood_point_found* then
10: $Count++$;
11: End if
12: End for
13: If Count < λN then
14: $Plane_j$ is a redundant facet
15: End if
16: End for
17: For eachn P_t in P_{ti} do
18: Neighborhood radius search in D
19: If *Neighborhood_point_NotFound* then
20: P_t is missing in the model
21: End if
22: End for

5.2. Model Colorization

The colorization of the model aims at situations where the 3D equipment model lacks textures. It involves aligning two point clouds and then coloring the corresponding facets of the model by querying the colors of the point cloud at the corresponding positions. The steps are outlined in Algorithm 2.

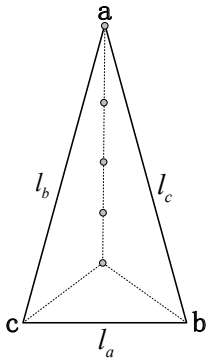
Similar to the model differencing method, first, point cloud P is transformed to point D's coordinate, resulting P_t. Then, based on the alignment of the model and the point cloud, the sampled points of the model fac-

ets are retrieved from the corresponding points in P_t.

First, a triangular facet $Plane_j$ is selected from M, as shown in Lines 4-9 of Algorithm 2. Next, the angle bisector line of the triangle is extracted. For this purpose, the incenter O of the triangle is computed based on the coordinates of its vertices. The lengths of the line segments connecting the incenter and the three vertices are computed. Among the three connecting lines, the longest line segment is selected as the angle bisector line. Then, five equidistant points are sampled along this line segment. A schematic diagram is shown in Figure 5. The incenter of the triangle is computed as shown in Equation (9), where a, b, and c represent the vertices of the triangle, and l_a, l_b, and l_c are the lengths of the corresponding edges.

$$O=\frac{l_a * a+l_b * b+l_c * c}{a+b+c} \tag{9}$$

Figure 5
Sampling the angle bisector line of a triangular facet



After the five points are sampled along the angle bisector line of $Plane_j$. For each of the five points, a nearest neighbor search is conducted in P_t to find 10 neighboring points. The resulting 50 points are recorded as a point set E. In Line 10-16, E is projected to $Plane_j$. For each candidate point p_i, we compute u and v according to Equation (10). If $0 \le u \le 1, 0 \le v \le 1, u+v \le 1$, point p_i is considered within the triangle and added to the point set F_i.

$$u=\frac{(\overrightarrow{ab}\cdot\overrightarrow{ab})*(\overrightarrow{ap_i}\cdot\overrightarrow{ac})-(\overrightarrow{ac}\cdot\overrightarrow{ab})*(\overrightarrow{ap_i}\cdot\overrightarrow{ab})}{(\overrightarrow{ac}\cdot\overrightarrow{ac})*(\overrightarrow{ab}\cdot\overrightarrow{ab})-(\overrightarrow{ac}\cdot\overrightarrow{ab})*(\overrightarrow{ab}\cdot\overrightarrow{ac})}$$
$$v=\frac{(\overrightarrow{ac}\cdot\overrightarrow{ac})*(\overrightarrow{ap_i}\cdot\overrightarrow{ab})-(\overrightarrow{ac}\cdot\overrightarrow{ab})*(\overrightarrow{ap_i}\cdot\overrightarrow{ac})}{(\overrightarrow{ac}\cdot\overrightarrow{ac})*(\overrightarrow{ab}\cdot\overrightarrow{ab})-(\overrightarrow{ac}\cdot\overrightarrow{ab})*(\overrightarrow{ab}\cdot\overrightarrow{ac})} \tag{10}$$

In Line 17, all points within the triangle are discovered and their average color is taken to colorize the triangular facet $Plane_j$. By doing this, the textless model M can be colorized according to the real-world observation of the equipment's point cloud.

Algorithm 2: Point Cloud-guided Model Colorization

Input: M: Textureless equipment model; P: Corresponding point cloud; r: search radius;
Output: M with textures

1: $D = PoissonDisk(M)$ //Model discretization
2: $T = Align(D, P)$ //Transform P to align with D
3: $P_t = T * P$
4: For each $Plane_j$ in M do
5: Compute the center of the triangle O
6: Compute the angle bisector line
7: Sample five points on the angle bisector line to form a set A
8: For all points in A do
9: Neighborhood radius search in P_t to obtain the point set E
10: End if
11: Project E to $Plane_j$
12: For each projected point p_i do
13: $u=\frac{(\overrightarrow{ab}\cdot\overrightarrow{ab})*(\overrightarrow{ap_i}\cdot\overrightarrow{ac})-(\overrightarrow{ac}\cdot\overrightarrow{ab})*(\overrightarrow{ap_i}\cdot\overrightarrow{ab})}{(\overrightarrow{ac}\cdot\overrightarrow{ac})*(\overrightarrow{ab}\cdot\overrightarrow{ab})-(\overrightarrow{ac}\cdot\overrightarrow{ab})*(\overrightarrow{ab}\cdot\overrightarrow{ac})}$
$v=\frac{(\overrightarrow{ac}\cdot\overrightarrow{ac})*(\overrightarrow{ap_i}\cdot\overrightarrow{ab})-(\overrightarrow{ac}\cdot\overrightarrow{ab})*(\overrightarrow{ap_i}\cdot\overrightarrow{ac})}{(\overrightarrow{ac}\cdot\overrightarrow{ac})*(\overrightarrow{ab}\cdot\overrightarrow{ab})-(\overrightarrow{ac}\cdot\overrightarrow{ab})*(\overrightarrow{ab}\cdot\overrightarrow{ac})}$
14: If $0 \le u \le 1, 0 \le v \le 1, u+v \le 1$ then
15: Add p_i to the point set F_i
16: End if
17: End for
18: Use the average color of the points in F_i to render $Plane_j$
19: End for

6. Experiments

6.1. Data Preparation for Training GKANet

We collected the LiDAR point cloud data in a 500KV power substation in Nanjing, China. An equipment

model library including 400 models such as lightning arrester, circuit breaker, gantry and electric reactor were also collected (as shown in Figure 6) from the digital twin of the substation. To perform the cross-source point cloud registration, the 3D models of the electrical equipment were resized for normalization. The 3D models were further discretized using grid sampling based on Poisson-disk distribution, as shown in Figure 7.

Figure 6
Equipment DT model library

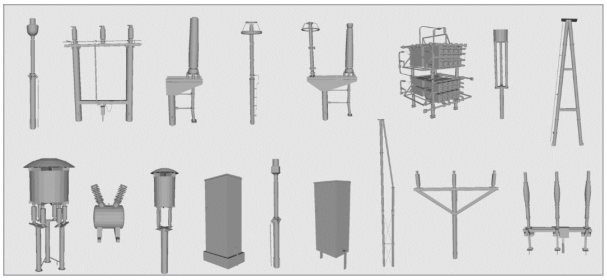
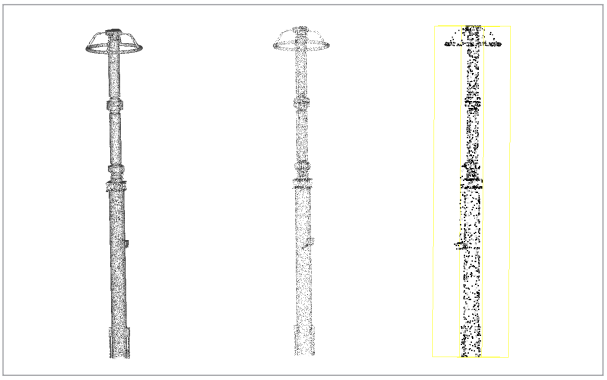


Figure 7
Discretization of power equipment 3D model



To improve the robustness of the network against the geometrical shape variations of the source and target electrical equipment, we developed a series of data augmentation operations for generating synthetic point clouds. Firstly, Gaussian noise was added to the point cloud to simulate LIDAR sensor noises. Then, some small parts were deleted from the point cloud at random positions to simulate occlusion and incomplete LIDAR scanning, which are common in practice. The electrical equipment 3D point cloud after random deletion is shown in Figure 8. Additionally, we also generated some small parts of point clouds with other objects and added them to the point cloud at random positions, as shown in Figure 9. This step takes into account the shape variance of real-world electrical equipment instances. Finally, rotations and translations were applied to the point cloud to simulate the device's different positions and orientations in the scene. In the end, a total of 10400 point cloud samples were obtained.

Figure 8
Models with random deletion of small parts

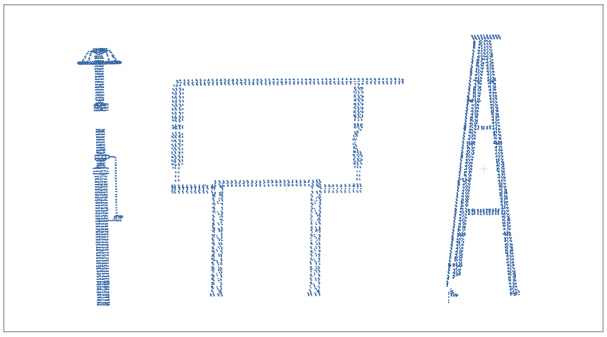
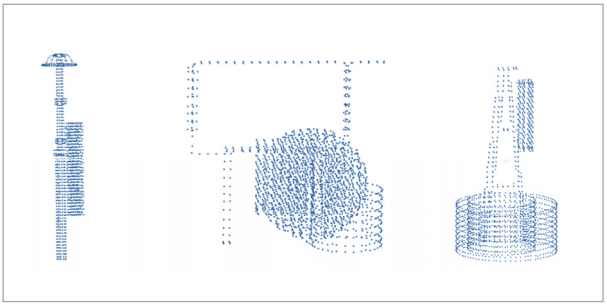


Figure 9
Models with random addition of small parts



6.2. Registration Results

The experiment was performed on a computer with Intel(R) i7 7700K and NVIDIA GTX 2080Ti. In the training of the network, we use SGD as the optimizer, and we set momentum=0.98, learning rate=0.01, ExpLR as the scheduler with $\lambda = 0.99$.

During the testing phase, we used real-world point cloud data as the source data and point clouds obtained by sampling the 3D models as the target data. The transformation matrix obtained through manual alignment was considered as the ground truth. Isotropic Rotation Error (IRE) and Isotropic Translation Error (ITE) [30] were selected as the evaluation metrics.

We further define those pairwise samples with IRE < 5°and ITE < 0.5m as a valid sample. Therefore, we can define mean IRE, mean ITE, and Precision as follows:

$$\text{RotationError}_{Mean} = \frac{1}{N_c}\sum_{i\in C}\arccos\left(\frac{\text{trace}(\mathbf{R}_i^T(\mathbf{R}_i^{GT})^{-1})-1}{2}\right), \quad (11)$$

$$\text{TranslationError}_{Mean} = \frac{1}{N_c}\sum_{i\in C}\|\mathbf{t}_i-\mathbf{t}_i^{GT}\|_2, \quad (12)$$

$$\text{Precision} = \frac{\mathbf{N}_C}{\mathbf{N}_T}\times 100\%, \quad (13)$$

where $\mathbf{N}_T$ denotes the total number of test samples, $\mathbf{N}_C$ is the number of valid test samples, C represents the set of valid samples, R_i and R_t respectively denotes the predicted rotation matrix and translation vector of the i th test sample, and $\mathbf{R}_i^{GT}$ and $\mathbf{t}_i^{GT}$ represent the ground truth, respectively. Note that our source and target point clouds were not identical, and thus both shape variation and the registration error can be the cause of the reprojection error. Therefore, the classic RMSE metric is not suitable for evaluating the registration accuracy in this case.

Firstly, the effectiveness of our data augmentation process was tested and the registration results are shown in Figure 10 and Table 1. Our registration network was trained on synthetic point clouds obtained from model discretization with or without data augmentation, and tested on the point clouds of real-world electrical equipment. However, most existing works train and test the registration network in one source of data. Therefore, our registration accuracy is generally lower than other similar works, e.g., [16] and [30]. The results in Figure 10 and Table 1 indicate that the point cloud augmentation can generate training data that is similar to real-world cases, thereby improving the sim-to-real generalization capability of the point cloud registration network.

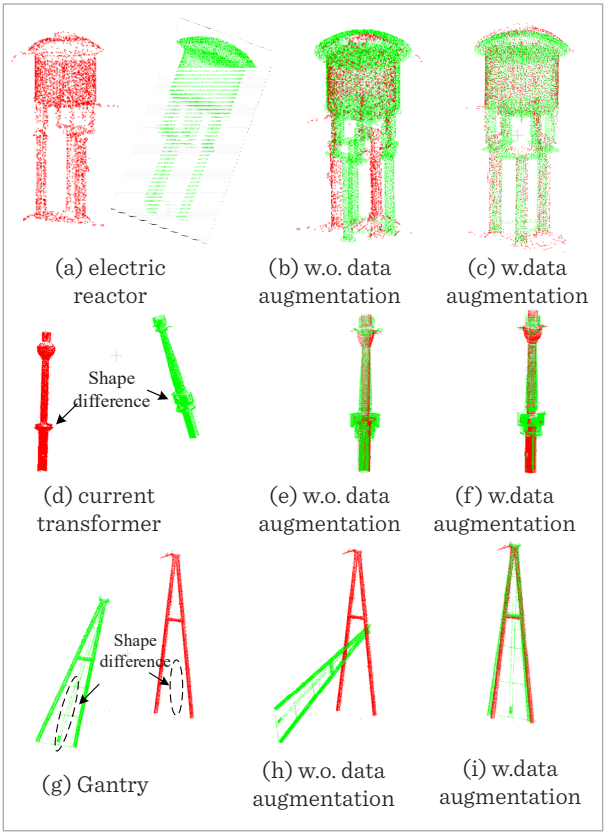
Table 1

Registration accuracy with or without data augmentation during training

Training Method	Rot. Error(°)	Trans. Error (m)	Precision (%)
w. o. DataAugment	5.370	0.889	68.9%
w. DataAugment	1.236	0.106	81.5%

Figure 10

Comparison of point cloud registration with or without data augmentation



(a) electric reactor (b) w.o. data augmentation (c) w.data augmentation

(d) current transformer (e) w.o. data augmentation (f) w.data augmentation

(g) Gantry (h) w.o. data augmentation (i) w.data augmentation

Then we compared the accuracy of the improved registration network with ICP [22] and the SOTA deep point cloud registration network Predator [6]. The registration results are shown in Figure 11 and the registration accuracy is shown in Tables 2-3. "SAC-IA+ICP" in the second line in Tables 2-3 indicate the SAC-IA algorithm based on the SHOT descriptor for coarse registration and ICP for fine registration. The first row of Figure 11 demonstrates an example of

Table 2

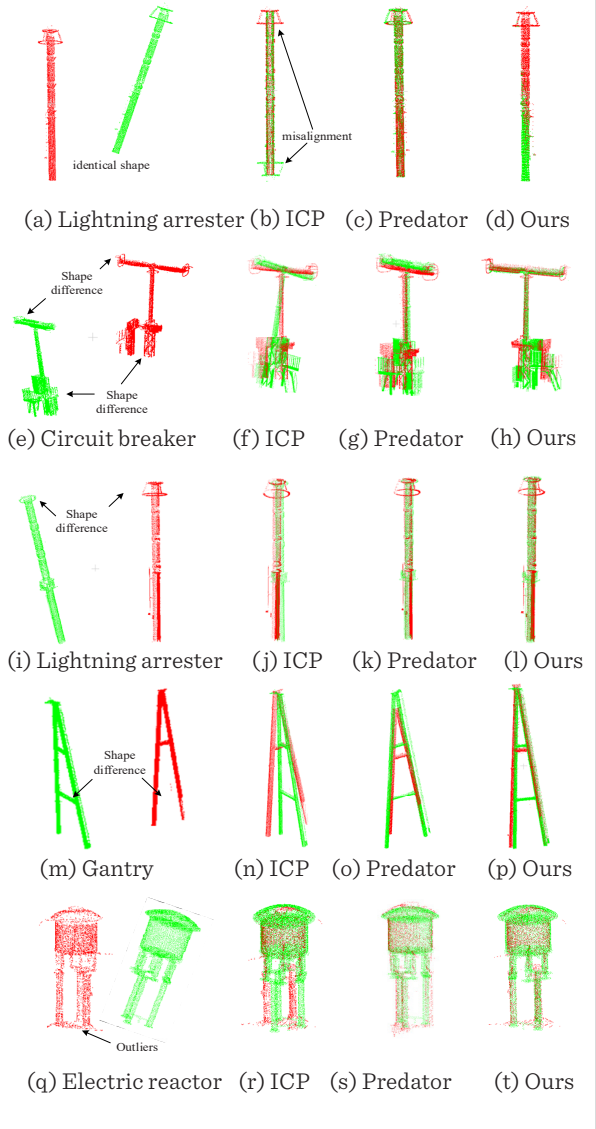
Accuracy of point cloud registration when pairwise objects are identical

Training Method	Rot. Error(°)
ICP	1.358e-02
SAC-IA+ICP	1.276e-04
Predator	1.557e-04
GKANet	1.039e-04

Table 3
Accuracy of point cloud registration when pairwise objects are with shape deviation

Training Method	Rot. Error (°)	Trans. Error (m)	Precision (%)
ICP	5.754	0.316	51.0%
SAC-IA+ICP	1.653	0.228	70.5%
Predator	1.515	0.186	73.2%
GKANet	1.236	0.106	81.5%

Figure 11
A comparison with other point cloud registration methods



an identical shape between the model and the point cloud, while the last four rows indicate the situation with shape deviation.

When the source equipment and target equipment are of identical shape (as shown in the first row of Figure 11), the accuracy shown in Table 2 indicates that all four methods achieve registration high accuracy. ICP algorithm occasionally fails to align symmetric objects such as the lightning arrester, as shown in Figure 11(b), and thus its accuracy is higher than the other three methods. We employ traditional RMSE metric for this case:

$$\text{RegRMSE} = \left(\frac{1}{N}\sum_{i=1}^{N} \| p_i - Rq_i - T \|\right)^{1/2}, \tag{14}$$

where N is the number of reference point, p_i denotes for a target point, and q_i represents a nearest point corresponding to p_i in the source point cloud.

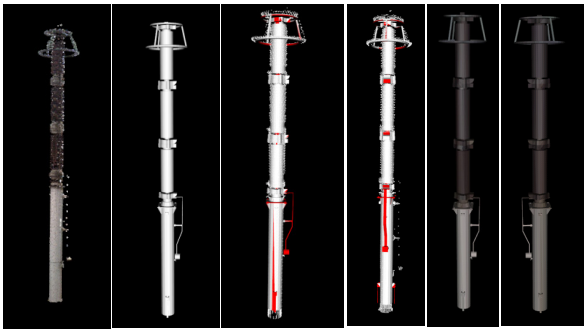
But when the paired model and point cloud belong to the same equipment category but have local geometrical shape deviation, the point cloud registration errors of all four methods are generally higher, as illustrated in Table 3. Figure 10(d)(g) and Figure 11 (e)(i)(m) depict examples of such deviations. As shown in Figure 11(q), outliers that do not belong to equipment will also bring about registration errors. In addition, the electrical equipment is generally larger than daily objects. That's why the errors in our experiment seem higher than in other similar studies. Nevertheless, experiment results validate that our improved registration network (with data augmentation) outperforms the ICP algorithm and Predator in dealing with real-world electrical equipment point clouds under the condition of shape deviation and noise. Note that in this case, we use the mean IRE and mean ITE metric. The reason is that the traditional RMSE metric will contain the reprojection error caused by the shape variance. Therefore, the RMSE metric is only taken as the evaluation metric of identical shape cases.

6.3. Model Verification Results

Figure 12 presents an example of the model verification results. Figure 12(a) depicts the point cloud of a lightning arrester extracted from the actual scene, with colored texture.

Figure 12(b) displays the textureless 3D model of the lightning arrester. The two have roughly similar shapes but deviations in some local shape details.

Figure 12
Model differentiation and colorization results



(a) (b) (c) (d) (e) (f)

Based on their alignment, Figure 12(c) and (d) illustrate the difference between the model and the point cloud, with the discrepancies automatically highlighted in red. Figure 12(e) and (f) demonstrate the automatic coloring of the textureless 3D model according to the observed texture of the point cloud in Figure 12(a). It is evident that the proposed method in this paper enables the verification of the shape differences between the 3D model and the real-world electrical equipment point cloud, as well as the correction of color textures.

7. Conclusions

In this paper, an equipment-level digital twin model verification method based on overlap-aware cross-source point cloud registration is presented. The key of the method is GKANet, which introduces grouped KPConv with an attention mechanism, as well as an overlap-weighted circle loss. Therefore, the accuracy of complex-shaped electrical equipment registration is improved, even under the condition of local geometric shape variations between a DT model and the corresponding LiDAR point cloud. Based on the pose alignment, model differentiation and colorization is achieved via computing the facet-point correspondence. The results demonstrated the proposed method's capability in automated shape and texture verification for electrical equipment. In the future, we plan to apply the point cloud registration method to the equipment in power transmission lines, in which the more challenging problem of scene registration will be addressed.

Acknowledgement

This work is supported by the Science and Technology Project of State Grid Corporation of China "Fast construction and collaborative analysis of digital twin of electric distribution grid based on integrated electrical topology" (5108-202218280A-2-396-XG).

References

1. Aoki, Y., Goforth, H., Srivatsan, R. A., Lucey, S. PointNetLK: Robust & Efficient Point Cloud Registration Using PointNet. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, Long Beach, CA, USA: IEEE, 7156-7165. https://doi.org/10.1109/CVPR.2019.00733
2. Bai, X., Luo, Z., Zhou, L., Fu, H., Quan, L., Tai, C.-L. D3Feat: Joint Learning of Dense Detection and Description of 3D Local Features. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, Seattle, WA, USA: IEEE, 6358-6366. https://doi.org/10.1109/CVPR42600.2020.00639
3. Han, G., Zhu, Y., Liao, L., Yao, H., Zhao, Z., Zheng, Q. Hybrid Attention-Based 3D Object Detection with Differential Point Clouds. Electronics, 2022, 11(23), 4010. https://doi.org/10.3390/electronics11234010
4. He, Y., Yang, J., Hou, X., Pang, S., Chen, J. ICP Registration with DCA Descriptor for 3D Point Clouds. Optics Express, 2021, 29(13), 946, 20423-20439. https://doi.org/10.1364/OE.425622
5. Hu, P., Ho, E. S. L., Munteanu, A. AlignBodyNet: Deep Learning-Based Alignment of Non-Overlapping Partial Body Point Clouds from a Single Depth Camera. IEEE Transactions on Instrumentation and Measurement, 2023,72,1-9.https://doi.org/10.1109/TIM.2022.3222501
6. Huang, S., Gojcic, Z., Usvyatsov, M., Wieser, A., Schindler, K. PREDATOR: Registration of 3D Point Clouds with Low Overlap. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, Nashville, TN, USA, 4265-4274. https://doi.org/10.1109/CVPR46437.2021.00425
7. Huang, X., Mei, G., Zhang, J. Cross-Source Point Cloud Registration: Challenges, Progress, and Prospects. Neurocomputing, 2023, 548, 126383. https://doi.org/10.1016/j.neucom.2023.126383

8. Jiang, F., Liu, Y., Chen, Z., Chen, Z. SourcePoint Cloud Registration of Vehicle-Mounted Lidar Based on Improved FPFH-ICP Algorithm. Nuclear Techniques, 2023, 46(5), 1-10. DOI: 10.11835/j.issn.1000-582X.2022.105
9. Lamine Tazir, M., Gokhool, T., Checchin, P., Malaterre, L., Trassoudaine, L. CICP: Cluster Iterative Closest Point for Sparse-Dense Point Cloud Registration. Robotics and Autonomous Systems, 2018, 108, 66-86. https://doi.org/10.1016/j.robot.2018.07.003
10. Liu, Q., Liu, Y., Yan, Y., Deng, J., Jiang, Q., Jiang, X. Hierarchical Cross-Source Point Cloud Registration Method for Power Equipment. High Voltage Engineering, 2022, 48(8), 2961-2971. DOI: 10.13336/j.1003-6520.hve.20220077
11. Liu, Y., Shao, Z., Hoffmann, N. Global Attention Mechanism: Retain Information to Enhance Channel-Spatial Interactions. arXiv:2112.05561, 2021.
12. Mellado, N., Aiger, D., Mitra, N. J. Super 4PCS Fast Global Point Cloud Registration Via Smart Indexing. Proceedings of the Symposium on Geometry Processing, 2014, 205-215. https://doi.org/10.1111/cgf.12446
13. Qin, Z., Yu, H., Wang, C., Guo, Y., Peng, Y., Ilic, S. GeoTransformer: Fast and Robust Point Cloud Registration with Geometric Transformer. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023, 1-16. https://doi.org/10.1109/TPAMI.2023.3259038
14. Ran, Y., Xu, X. Point Cloud Registration Method Based on SIFT and Geometry Feature. Optik, 2020, 203, Art. no. 163902. https://doi.org/10.1016/j.ijleo.2019.163902
15. Sarrode, V., Li, X., Goforth, H., Aoki, Y., Srivastan, R. A., Lucey, S., Choset, H. PCRNet: Point Cloud Registration Network Using PointNet Encoding. arXiv, 2019. http://arxiv.org/abs/1908.07906
16. Shen, X., Xu, Z. A Multitemporal Point Cloud Registration Method for Evaluation of Power Equipment Geometric Shape. IEEE Transactions on Instrumentation and Measurement, 2022, 71, 1-14. https://doi.org/10.1109/TIM.2022.3203460
17. Shi, K., Qian, K., Yu, H. Visual Human Localization and Safety Monitoring in a Digital Twin of Workspace. 2022 41st Chinese Control Conference (CCC), Hefei, China, 2022, 6117-6121. https://doi.org/10.23919/CCC55666.2022.9902056
18. Sun, Q., Xu, Y., Sun, Y., Yao, C., Lee, J. S. A., Chen, K. GN-CNN: A Point Cloud Analysis Method for Metaverse Applications. Electronics, 2023, 12(2), 273. https://doi.org/10.3390/electronics12020273
19. Sun, W., Liu, L., Ji, X., Sun, C. Indoor LiDAR 3D Mapping Algorithm with Semantic-Based Registration and Optimization. Soft Computing, 2020, 24, 5909-5920. https://doi.org/10.1007/s00500-019-04482-6
20. Sun, Y., Cheng, C., Zhang, Y., Zhang, C., Zheng, L., Wang, Z. Circle Loss: A Unified Perspective of Pair Similarity Optimization. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA: IEEE, 2020, 6397-6406. https://doi.org/10.1109/CVPR42600.2020.00643
21. Thomas, H., Qi, C. R., Deschaud, J. E., MArcotegui, B., Goulette, F., Guibas, L. J. KPConv: Flexible and Deformable Convolution for Point Clouds. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South): IEEE, 6410-6419. https://doi.org/10.1109/ICCV.2019.00651
22. Wang, Q., Tan, Y., Mei, Z. Computational Methods of Acquisition and Processing of 3D Point Cloud Data for Construction Applications. Archives of Computational Methods in Engineering, 2020, 27, 479-499. https://doi.org/10.1007/s11831-019-09320-4
23. Wang, Y., Sun, Y., Liu, Z. Dynamic Graph CNN for Learning on Point Clouds. ACM Transactions on Graphics, 2019, 38(5), 1-12. https://doi.org/10.1145/3326362
24. Wu, Y., Yao, Q., Fan, X., Gong, M., Ma, W., Miao, Q. PANet: A Point-Attention Based Multi-Scale Feature Fusion Network for Point Cloud Registration. IEEE Transactions on Instrumentation and Measurement, 2023, 72, 1-13. https://doi.org/10.1109/TIM.2023.3271757
25. Xiao, P., Zhao, R., Li, D., Zeng, Z., Qi, S., Yang, X. As-Built Inventory and Deformation Analysis of a High Rockfill Dam Under Construction with Terrestrial Laser Scanning. Sensors, 2022, 22(2), 521. https://doi.org/10.3390/s22020521
26. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K. Aggregated Residual Transformations for Deep Neural Networks. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, 5987-5995. https://doi.org/10.1109/CVPR.2017.634
27. Yang, H., Shi, J., Carlone, L. TEASER: Fast and Certifiable Point Cloud Registration. IEEE Transactions on Robotics, 2021, 37(2), 314-333. https://doi.org/10.1109/TRO.2020.3033695
28. Yang, Z., Qin, Y., Wang, X., Peters, R. Automated Registration of Cross-Source and Multi-Temporal Point Clouds in Urban Areas. Chinese Journal of Lasers, 2023, 50(10). https://doi.org/10.3788/CJL221360
29. Yang, Z., Wang, X., Hou, J. A 4PCS Coarse Registration Algorithm Based on ISS Feature Points. Chinese Con-

trol Conference (CCC), 2021-July, 7371-7375. https://doi.org/10.23919/CCC52363.2021.9549486

30. Yew, Z. J., Lee, G. H. RPM-Net: Robust Point Matching Using Learned Features. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 11821-11830. https://doi.org/10.1109/CVPR42600.2020.01184
31. Yuan, Y., Ge, Z., Lai, B., Guo, X., Zhang, Y., Liu, X., Suo, T., Yu, Q. Three-Dimensional Deformation Measurement Method Based on Image-Guided Point Cloud Registration. Optics and Lasers in Engineering, 2023, 161, 107399. https://doi.org/10.1016/j.optlaseng.2022.107399
32. Zhang, K., Hao, M., Chen, X., Leng, Y., et al. Linked Dynamic Graph CNN: Learning Through Point Cloud by Linking Hierarchical Features. 2021 27th International Conference on Mechatronics and Machine Vision in Practice (M2VIP), Shanghai, China, 2021, 7-12. https://doi.org/10.1109/M2VIP49856.2021.9665104
33. Zhao, Y., Zhu, Z., Liu, W., Zhan, J., Wu, D. Application of 3D Laser Scanning on NATM Tunnel Deformation Measurement During Construction. Acta Geotechnica, 2023, 18, 483-494. https://doi.org/10.1007/s11440-022-01546-0
34. Zheng, L., Li, Z. Virtual Namesake Point Multi-Source Point Cloud Data Fusion Based on FPFH Feature Difference. Sensors, 2021, 21(16). https://doi.org/10.3390/s21165441