

Research Article

Point Cloud Key Point Extraction Algorithm Based on Feature Space Value Filtering

Guowei Chen 

New Media Institute, Communication University of China, ChaoYang, Beijing 100024, China

Correspondence should be addressed to Guowei Chen; 201772442@yangtzeu.edu.cn

Received 22 June 2022; Revised 7 August 2022; Accepted 10 August 2022; Published 28 August 2022

Academic Editor: Muhammad Zakarya

Copyright © 2022 Guowei Chen. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the rapid development of 3-dimensional (3D) acquisition technology, point clouds have a wide range of application prospects in the fields of computer vision, autonomous driving, and robotics. Point cloud data is widely used in many 3D scenes, and deep learning has become a mainstream research method for classification with the advantages of automatic feature extraction and strong generalization ability. In this paper, a hierarchical key point extraction framework is proposed to solve the problem of modeling the local geometric structure between points. Various point cloud models such as PointNet, PointNet++, and DGCNN are analyzed and their features in local key point are extracted. Based on these analyses, an indexed edge geometric feature spatial value screening neural network (IEGCNN) is proposed. This network extracts features from each point and its neighborhood, calculates the distance between the center point and the points within its neighborhood, and adds the point orientation information to the edge feature spatial value screening network. The relationship between points in the edge network architecture is projected onto a 3D coordinate system and decomposed into three orthogonal bases. The geometric structure between two points is modeled by feature aggregation based on the angle between the edge vector and the base vector and the distance between the center point and the neighboring points. The proposed method has the capability of fast processing of point cloud data by significantly reducing the training and recognition time. The experimental results show that this method achieved high classification accuracy value. This work also provides an idea to solve the problem of real-time target detection network, which has a broad applications prospect in the deployment of movable devices and real-time processing.

1. Introduction

Three-dimensional (3D) target recognition is a hot research topic nowadays, where the key to target recognition is target classification and segmentation [1–3]. With the development of data acquisition equipment, various means of data acquisition are becoming more and more abundant, from pictures to 3D models. Point cloud is the main expression line of 3D data; LIDAR, depth sensor, and other equipment can directly collect point cloud data [4–6]. Point clouds can be widely used in various fields of logistics, such as intelligent robots and unmanned vehicles. Intelligent robots use LIDAR to collect point cloud data. The traditional method uses video and other data, and its efficiency is low, while the point cloud data collected by intelligent robots has high efficiency, so the application of point cloud in the intelligent sorting

system of logistics storage can improve the efficiency of sorting. Driverless cars use depth sensors to collect point cloud data in the environment, and the analysis and processing of point cloud data for obstacle avoidance and environment perception can improve the accuracy and time efficiency of environment perception [7–10]. Therefore, more and more scholars are focusing their efforts on the processing of point cloud data. Point cloud classification is like image classification, and the implementation principle is to correctly identify point cloud data according to the corresponding labels. Point cloud segmentation is to categorize point cloud data according to rules, and usually points with the same features are labeled into a class. Traditional methods generally extract features manually, and the key points are extracted to classify and segment the point clouds, but such methods rely on the professional level and

experience of human, and the process is more complicated and only applicable to specific tasks. Currently, in order to improve the automation of point cloud classification and segmentation, deep learning is used instead of traditional methods. Deep learning can extract high-dimensional features of the input data according to the learning objective. In recent years, many researchers have applied deep learning techniques from 2D domain to 3D unstructured point cloud data and can directly deal with disordered and sparse point cloud data. However, most methods only extract global features of point clouds, ignoring the local relationships existing between points and points, resulting in low accuracy and robustness of classification segmentation. In addition, deep learning is computationally complex and memory consuming and requires a large amount of training time, which is not conducive to transferring deep learning models to practical application scenarios. Therefore, it is of strong practical significance to improve the accuracy of point cloud classification segmentation, reduce memory loss, and improve the efficiency of training time [11–15].

Point cloud data is a set of vector collections in a 3D coordinate system, a 3D model representation, and each point is usually represented by 3D coordinates. Compared with 2D images and grid images, point cloud data contains RGB values, grayscale values, and other information, which can visually depict the real world. Point cloud data is the most primitive data, which does not need preprocessing and can be directly applied to deep learning methods for processing and analysis. Voxel data, on the other hand, generally has to be preprocessed before deep learning methods can be used. Example diagram of an unordered point cloud is shown in Figure 1.

As can be seen from Figure 1, the point cloud data consists of discrete points and does not contain structured information, which cannot be directly processed by traditional methods. Point cloud data is different from other data, it is disordered, and the same point may have many different manifestations, which leads to the traditional deep learning that cannot process the point cloud data, and the point cloud data has certain sparsity, and it is difficult to use the deep learning method to process the sparse point cloud when the collection device collects the point cloud data. As shown in the figure, the point cloud consists of four points, f_a , f_b , f_c , and f_d , and the point cloud data collected by different devices may have different orders, and the traditional methods will incorrectly identify the point clouds with different orders as different classes [16–19].

With the widespread use of many sensor devices and the increasing development of information technology, people's daily life and many fields such as engineering and technology generate various forms of massive image data, and the technology used to obtain "information" from the massive image data and guide machines to perceive and understand their surroundings is computer vision. Image alignment is an important step in many complex computer vision tasks, aiming to match multiple spatial data acquired by different sensors at different times and under different conditions, and is therefore one of the difficult and hot research areas in computer vision. Image alignment technology emerged in

the 1970s and was early applied to military weapons and equipment. After rapid development, image alignment has been gradually applied to many civilian fields and widely used in different disciplines and different research tasks. Image alignment has been widely used in many fields and in imaging medicine, which produces multimodal data with a high degree of variability. Image alignment aligns multimodal organ data to the same coordinate system, and through image fusion, it can reflect the tissue shape and function of the organ at the same time, facilitating medical diagnosis. In mapping, image alignment technology aligns high-resolution serial images collected by UAVs to the same spatial coordinate system, and further stitching and fusion can generate a panoramic map of the target scene. Image alignment algorithms are also required as the basis for different tasks, such as 3D reconstruction, simultaneous visual localization and mapping, image stitching and fusion, image retrieval, target identification and tracking, and other complex computer vision tasks. 3D reconstruction restores the 3D structure of a static scene by recovering it from images with different viewpoints in three main steps, feature matching between images, camera pose estimation, and recovery of 3D structure using estimated motion and features. Image alignment is the first step of image processing, and the accuracy of its algorithm has a great impact on the 3D reconstruction results. Image stitching and fusion is the process of stitching and aligning two or more images with overlapping scenes obtained from different viewpoints, different times, or different sensors to produce a large field of view image, which is the most direct application of image alignment algorithms. The goal of the image retrieval task is to retrieve all images with similar scenes in each query image, where the image alignment algorithm compares the similarity between images by computing feature matching between images. In summary, the significance of image alignment is to establish the correspondence between the two or more images to be aligned, the image and the target object, or the image and the features extracted by the template. In recent years, image alignment techniques have achieved some results and have shown broad application prospects in many vision tasks. As a key technology in many fields, the evolving computer vision has put forward higher requirements on the alignment technology, so the alignment technology still has important research significance and practical value [20].

Point cloud data is widely used in many 3D scenes, and deep learning has become a mainstream research method for point cloud classification. According to the different ways of feature space value screening, the existing algorithms are categorized into traditional methods and deep learning algorithms. In this paper, based on the representative methods and the latest research of deep learning, the basic ideas are summarized with its advantages and disadvantages; compare and analyze the experimental results of the main methods; look forward to the future work and research development direction of deep learning in the field of point cloud key point extraction. There are still problems of excessive number of parameters and complex network size when the original point cloud is directly input to the classification

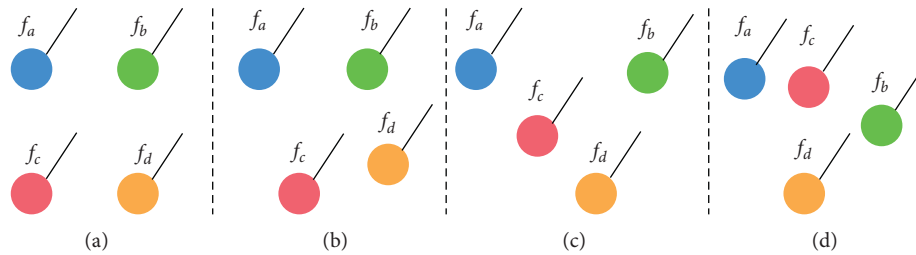


FIGURE 1: Diagram of a disordered point cloud.

network, and the real-time task processing still needs further optimization. The existing 3D point cloud key point extraction methods usually ignore the useful information in other neighborhood features, so this paper proposes a point cloud key point extraction algorithm based on feature space value screening. Firstly, the network structure and super-parameters are trimmed and compressed to achieve a lightweight model; secondly, the k -nearest neighbor algorithm is used to determine a new local region on each feature space value screening layer, add the vector direction between neighboring points, map the output features of different layers, and make index jump connections to further reduce the local feature information loss. This will have a broad applications prospect in the deployment of movable devices and real-time processing.

2. Related Work

This section will elaborate point cloud key point extraction mechanisms and point cloud alignment techniques.

2.1. Point Cloud Key Point Extraction. In recent years, with the continuous development of science and technology and the rapid growth of life needs, the digital modeling of 3D objects in the objective world with point cloud data, i.e., 3D reconstruction technology, has been frequently used in various industries and has received good response. In the field of medicine, through the construction of three-dimensional model of human organs, there can be more intuitive response to the patient's lesions, to help doctors develop more effective treatment plan; in the field of public transportation, there is autonomous driving technology by real-time construction of the vehicle's surrounding environment, to help the vehicle in time to find and avoid obstacles; in the field of archaeology, through the construction of three-dimensional model of cultural relics, it can digitally repair the broken cultural relics and restore the historical appearance; in the field of archaeology, by constructing 3D models of cultural relics, it can digitally restore damaged relics and restore the historical appearance. The most basic and critical step in the realization of 3D reconstruction is the point cloud key point extraction [21–25]. When using 3D scanning equipment to scan an object, it is impossible to obtain complete 3D information of the surface of the object in a single scan due to object occlusion, limited field of view, and other factors in a fixed perspective.

Therefore, the object needs to be scanned in multiple views, and then the results of multiple scans are stitched together and fused, and the fusion of point cloud data from two adjacent views uses the point cloud key point extraction technique. By solving the spatial transformation relationship between two coordinate systems, a rotation translation matrix can be obtained, and then the rotation translation matrix converts the two point clouds into the same coordinate system to realize the point cloud summary key point extraction, and so on to realize the 3D reconstruction of the object. However, in practice, due to the accuracy of 3D scanner measurement, human operation, or the influence of natural environment, the obtained point cloud data have errors, which will affect the key point extraction effect of two adjacent point clouds and thus affect the final 3D reconstruction effect. Therefore, this paper will start from the point cloud key point extraction algorithm, analyze several common point cloud key point extraction algorithms, and propose improvements, aiming to design a point cloud key point extraction algorithm that takes into account efficiency, accuracy, and robustness and provide a good technical support for the development of subsequent point cloud key point extraction technology and 3D reconstruction technology [26].

The point cloud key point extraction algorithm based on deep learning is based on inputting a point in the point cloud data and its neighborhood information into a neural network and describing the point using the output vector of a layer in the network. 3D ShapeNets lead deep learning to 3D modeling and extract global features by calculating deep key points of 3D data with good noise immunity, but poor extraction of key points for low overlap rate point cloud models. 2D feature space value screening neural network is used to generate descriptors for local feature matching, but it only connects image block feature vectors as training samples for the network, thus lacking spatial correlation. 3DMatch algorithm, a self-supervised learning method, uses millions of positive and negative labels in RGB-D reconstruction results to train robust descriptors by twin neural networks for point cloud key point extraction. Based on 3DMatch, KNN is used to find the corresponding points to improve the efficiency of the algorithm. Based on the 3DMatch network framework, more descriptive and distinguishable descriptors are trained by increasing the number of feature space value screening layers and eliminating pooling layers, but the training efficiency is low. Perfect-Match algorithm vowelizes the network input for

density values, reduces the voxel grid density, and saves the network capacity, and the algorithm has real-time performance. The binarized local feature descriptors can effectively reduce the computational effort. By increasing the negative sample weights, a multiedge-based loss function is proposed, which increases the gap between positive and negative samples. A 3D point cloud-based self-coding descriptor (Adaptive O-CNN) is proposed, which can retain more information of the original point cloud. Point cloud key point extraction plays a central role in 3D scene reconstruction, which is essentially to find the transformation relationship between the point cloud data to be matched. In early studies on point cloud key point extraction, key points with distinguishing and descriptive power are usually pre-calculated for point cloud data and then further processed. However, recent studies have shown that the existing key point detection methods are not only time-consuming but also ineffective in practical applications, while uniform sampling and random sampling methods are proved to be an effective method to replace the key point detection algorithm. Therefore, the research direction of feature-based point cloud key point extraction is mainly directed at feature description and feature matching. Table 1 collates the common point cloud feature description methods.

2.2. Point Cloud Alignment. In this section, point cloud alignment techniques are divided into two main categories: alignment based on traditional methods and alignment based on deep learning methods, and an overview of these two types of point cloud alignment methods is given. In traditional methods, traditional point cloud alignment calculates the spatial transformation relationship between two point clouds through spatial geometry theory or statistical principles, which can be divided into two categories: feature-based alignment algorithms and featureless-based alignment algorithms [27–29].

Feature-based alignment algorithms mainly use the feature points of point clouds to achieve point cloud alignment. The feature points are the rotation invariant points in the point cloud data, and the local space is constructed by taking the feature points as the center, extracting the information in the space to describe the feature points, judging whether they are the corresponding points by the similarity of the two point descriptions, and calculating the spatial transformation matrix between the two point clouds according to the corresponding point set to realize the point cloud alignment. The Harris algorithm is suitable for gridded point cloud data, which needs to be gridded before feature detection, so it is not real-time; the intrinsic shape signature algorithm defines a local coordinate system for each point in the point cloud and establishes the covariance matrix and solves it to determine the feature points according to the relationship between the magnitude of the feature values; the SIFT algorithm considers the extreme values of the adjacent scales of key points. The NARF algorithm only targets the edge feature points, so the detected feature points have limitations and are sensitive to noise and outliers. There are also many methods for feature descriptor construction. The

concept of point signature, which generates a point signature for each point in the point cloud and finds the corresponding point pair by judging the similarity of the signature, is computationally intensive. The point feature histogram descriptor needs to calculate the distance and angle relationships of all points in the neighborhood of the feature points, and thus the algorithm is inefficient. Further, the fast point feature histogram descriptor, which simplifies some feature components in PFH, reduces the computational effort while maintaining the descriptiveness of the descriptor as much as possible [30]. By constructing a local reference system of feature points, dividing regions based on the interval and direction of points and feature points in the neighborhood of feature points, calculating the normal vectors of feature points in the regions and their angle cosines, and performing histogram statistics for each region, the histogram obtained by connecting each region is finally used as the SHOT descriptor. The matching of descriptors is accelerated by binarizing SHOT descriptors. Feature descriptors with scale invariance enable point cloud alignment at each scale. RIFT descriptors have rotation invariance. The residual angle in the adaptive domain is used as a feature descriptor of points for point cloud alignment at scaling scales. Various local features are combined such as normal and density as feature descriptions of points, which is computationally low and descriptive [38, 39].

Featureless-based alignment algorithm method refers to the direct manipulation of the original point cloud data to achieve alignment. The iterative nearest point algorithm has certain requirements on the initial positions of the two point clouds to be aligned, and when the initial positions of the two point clouds are far apart, the alignment is not effective or even cannot be achieved, and it is sensitive to extraneous factors such as noise. Using the point-to-point tangent plane distance instead of the point-to-point distance in the original ICP algorithm, the efficiency is improved by reducing the number of iterations. The ICP algorithm based on K - D tree accelerated search improves the efficiency of the algorithm. Point-to-point and point-to-surface feature definitions reduce the sensitivity of the ICP algorithm to the initial position of the model by estimating the curvature of all points in the point cloud but are not as efficient. Constrained ICP algorithm, by dividing the space and reducing the corresponding point search space, improves the efficiency of the algorithm. A search method based on PD-tree structure is robust to noise. The Pickey-ICP algorithm uses the idea of hierarchy to improve the efficiency of searching corresponding point pairs. The GO-ICP algorithm effectively solves the problem that the ICP algorithm easily falls into local optimal solutions. The ICP algorithm incorporating genetic algorithm takes the result of alignment using genetic algorithm as the initial position of the point cloud to improve the alignment accuracy. The ICP algorithm based on curvature extrema accelerates the convergence of the algorithm by constraining the curvature extrema. Hu improves the alignment accuracy of ICP algorithm by using dynamic angle factor. 4-point matching algorithm, however, generates incorrect alignment when the point cloud model has symmetry. Based on 4PCS, the Super4PCS algorithm is

TABLE 1: Point cloud key point extraction method.

Algorithm name	Algorithm categories
3DSC [31]	Point cloud descriptors based on local features
RoPS [32]	Point cloud descriptors based on local features
SHOT [33]	Local feature-based point cloud descriptor based on local features
PFH [34]	Local feature-based point cloud descriptor based on local features
FPFH [35]	Local feature-based point cloud descriptor
PointNet [36]	Convolutional neural network-based point cloud descriptor based on convolutional neural network
3DSmoothNet [37]	Convolutional neural network-based point cloud descriptor based on convolutional neural network

proposed to provide alignment efficiency while ensuring alignment accuracy. The chunked point cloud variance distribution similarity principle is used to extract the point cloud overlap region, which improves the accuracy of point cloud alignment with low overlap rate. The algorithm accuracy is improved by constructing an objective function to reduce the error accumulation.

The existing 3D point cloud key point extraction methods usually ignore the useful information in other neighborhood features, so this paper proposes a point cloud key point extraction algorithm based on feature space value screening. Firstly, the network structure and super-parameters are trimmed and compressed to achieve a lightweight model; secondly, the k -nearest neighbor algorithm is used to determine a new local region on each feature space value screening layer, add the vector direction between neighboring points, map the output features of different layers, and make index jump connections to further reduce the local feature information loss. This will have a broad applications prospect in the deployment of movable devices and real-time processing.

3. Methods

In this section, the proposed system is discussed in detail. The proposed network structure is presented and the edge geometry feature space value filtering is discussed.

3.1. Model Architecture. Based on the principle of network light-weighting, a prototype network structure is proposed based on PointNet as shown in Figure 2. By simplifying the network structure, only the basic feature space value filtering layer, the pooling layer, and the fully connected layer are included in the network to achieve a lightweight network. In order to extract the global features of the point cloud, the maximum pooling layer is used to extract the key points, and the size of the feature space value filtering kernel is set to 1×1 . Since each point in the fully connected layer is connected to all points in the previous layer, which integrates the features of the previous layers, the number of parameters in the fully connected layer is the largest in the whole network architecture, and the streamlining of the number of parameters and nodes in the fully connected layer is an important step to realize the network light-weighting. In the process of network optimization, the parameters of other layers can be kept consistent with the network prototype in order to investigate the impact of the parameters of one layer on the network performance.

3.2. Point Cloud Key Point Extraction. Traditional point cloud key point extraction methods are usually designed to solve domain-specific problems, and it is difficult to extend to new key point extraction tasks. Deep learning-based point cloud key point extraction can be divided into point-based extraction methods and tree-based extraction methods. The former directly uses the original point cloud as the input for deep learning; the latter first uses a k -dimensional tree structure to regularize the point cloud and then provides the processed data to the deep learning model. A class of feature space value filtering operation called Geo-Conv is applied to each point and its local neighborhood to extract the edge features of the central point and adjacent points by gradually expanding the acceptance domain of the feature space value filtering to extract features in layers and maintain the geometric structure of points along the hierarchy. Considering the directional information between points, the value of point projection to polar coordinates is calculated and then weighted and summed with the distance between two points to solve the problem of incomplete extraction of local key points.

3.3. Feature Space Value Filtering Based on Edge Feature Space Value Filtering. The key point extraction based on edge feature space value filtering uses the k -nearest neighbor method to define the k points closest to a point as the neighboring area. Firstly, the edge features between the center point and the neighboring points are extracted, and then the feature space value filtering operation is performed on the edge features. The set of nearest neighboring points to the centroid P_{xi} is $\{j:(i, j) \in \mathcal{E}\}$, and the set of directed edges associated with it is $\{(i, j_{i1}), \dots, (i, j_{ik})\}$. The edge features are defined as e_{ij} , where h_{θ} is a nonlinear function composed using the learnable parameter θ . An asymmetric aggregation operation Ψ is added to the h_{θ} operation to obtain the feature output of the i -th vertex of the edge feature space value screening:

$$P_{xi'} = \Psi_{j:(i,j) \in \mathcal{E}} h_{\theta}(P_{xi}, P_{xj}). \quad (1)$$

For the features of the centroid, the feature difference between the centroid and the neighboring points are fed into the multilayer perceptron in series, so that the edge features fuse the local relationship between points and the global information of the points. After obtaining n edge features, maximum pooling is performed to obtain a single feature of this local region, and local information is extracted and integrated layer by layer by superimposing multiple layers of

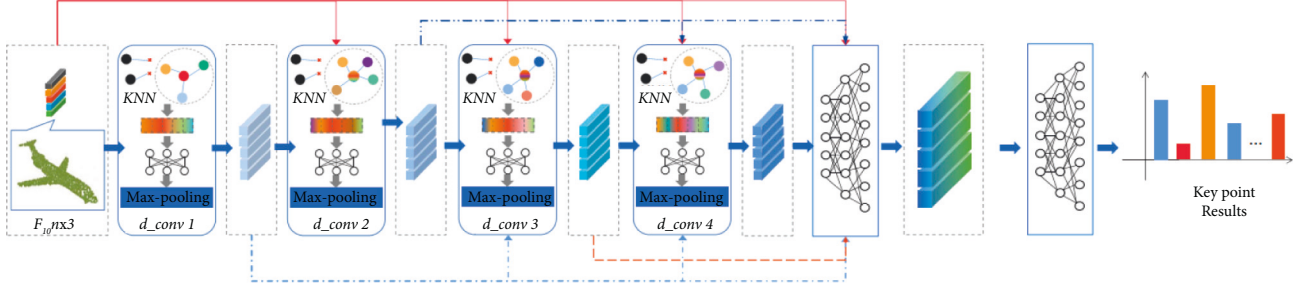


FIGURE 2: Proposed model structure.

feature space value screening in this way. The local neighborhood map of the edge feature space value filtering layer is constructed by a multilayer perceptron. When the edges of adjacent points are filtered by layer-by-layer eigenspace values, each layer outputs a new point cloud map structure and feature space, and a new local area is obtained. The method of interpoint difference is introduced to consider the geometric correlation information between points and solve the problem of incomplete extraction of local key points in PointNet and PointNet++ architectures; however, there is a problem of ignoring the directional information of points, so the indexed edge geometry feature space value screening neural network is proposed.

3.4. Key Point Extraction of Indexed Edge Geometry Feature Space. The edge geometry feature space value filtering is shown in Figure 3. Indexed edge geometry feature space value filtering neural network adds the orientation information of points to the edge feature space value filtering network, models the 3D point cloud with polar coordinate system, and projects the relationship between points in the edge network architecture to the 3D coordinate system. The values of polar coordinate projections to different axes are calculated and compared with two points

$$\begin{cases} P_{lj} = \sqrt{x^2 + y^2 + z^2}, \\ P_d = \left(\sqrt{x^2 + y^2}, \sqrt{y^2 + z^2}, \sqrt{z^2 + x^2} \right), \\ P_\theta = \arccos \frac{P_d}{P_{lj}}, \end{cases} \quad (2)$$

where P_{lj} denotes the mode length of vector, P_d denotes the projection length, and P_θ denotes the pinch angle. Suppose an F -dimensional point cloud contains n points; in the F -dimensional point cloud, n denotes the number of points and F denotes the number of channels. For each point, according to the k -nearest neighbor algorithm and the network superparameter r to construct a local spherical neighborhood, a spherical neighborhood $N(P_{xi})$ with P_{xi} as the center point can be constructed, and the feature output P_{xi} of the center point is calculated after obtaining several neighborhood points and then the center point, where the dimension of the weight matrix MF is $\mathcal{O}_{in} \times \mathcal{O}_{out}$.

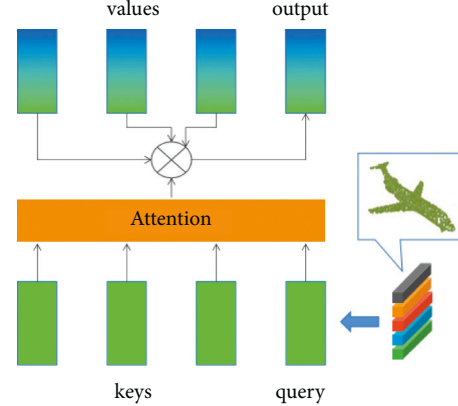


FIGURE 3: Edge geometry feature space value filtering.

$$N(P_{xi}) = \{P_{xj} | P_{xj} - P_{xi} \leq r\},$$

$$f(p_i | l) = \mathbf{M}_F f(p_i | (l-1)) + \frac{\sum_{i=1, j=1 (i \neq j)}^n w(P_{xi}, P_{xj}, r) h(P_{xi}, P_{xj}) P_{xi'}}{\sum_{i=1, j=1 (i \neq j)}^n w(P_{xi}, P_{xj}, r)} \quad (3)$$

where $\mathbf{f}(p_i | l)$ denotes the eigenvector of point p_i at layer l , \mathbf{M}_F denotes the weight matrix used to extract the features of the centroid, and w denotes the distance weighting between the centroid P_{xi} and different neighboring points P_{xj} , which decreases monotonically with $\|P_{xj} - P_{xi}\|$. As the radius r increases, the perceptual field of the spherical neighborhood gradually increases and the difference with the weight function $w(\cdot)$ of the center point decreases. $h(P_{xi}, P_{xj})$ denotes the edge feature, which is the most important feature extracted part of this network architecture. In the 3D Euclidean space, the vector can be expressed as the projection of three orthogonal bases, and the modulus of the projection represents the ‘‘energy’’ in the corresponding direction, so the edge features can be projected onto the three orthogonal bases, and the edge features in each direction are extracted using different weight matrices, and then the features in the three directions are regrouped to maintain the Euclidean geometry. The edge features based on polar coordinates are calculated as follows:

$$h(P_{xi}, P_{xj}) = \sum_{d \in D} \cos^2(P_\theta) M_d f(p_j | (l-1)), \quad (4)$$

D denotes the set of three orthogonal bases in the quadrant where P_{x_i} and P_{x_j} are located; M_d denotes the direction-dependent weight matrix to extract edge features in different directions; $\cos^2(P_\theta)$ is the coefficient to ensure that the sum is 1 when the features are aggregated.

3.5. Extraction of Key Points for Graph Feature Space Value Screening. Graph feature space value filtering is shown in Figure 4. In this paper, a directed graph $G=(V, E)$ is constructed by using the K -nearest neighbor operator, V represents the input point cloud, which is the vertex of the graph model, $V \in [1, N]$, N is the number of point clouds, E is the edge composed of point pairs, and the structure of KNN model is shown in Figure 4. x_i is the node; e_{ij} represents K neighboring points and node. The output of the graph model is the feature aggregation of all directed edges of the node, which is expressed formally as

$$x_i = \sum_{j:(i,j) \in E} h(x_i, x_j), \quad (5)$$

where $h(x_i, x_j)$ denotes the edge function, to reduce the number of parameters to improve the efficiency of the deep network while taking into account the local information of the point cloud; the edge function is defined to consider only local features:

$$h(x_i, x_j) = h(x_j - x_i). \quad (6)$$

The feature vector e_{ij} of node x_i is defined as

$$e_{ij} = g\left(w \sum_{i=1}^k x\right), \quad (7)$$

where w is the weight assigned to the node; g is the max pooling symmetric function, which is used to aggregate the feature vectors of the neighboring points.

3.6. Point Cloud Key Point Extraction Steps. The specific steps of point cloud key point extraction are as follows:

- (1) Input 7-dimensional point cloud with fused spectral and laser intensity information and solve the rotation invariance of the point cloud by aligning the input data with T-Net constructed by KNN.
- (2) Use MLP (64, 64) abstraction to align the shallow features of the point cloud and map each point feature to 64 dimensions.
- (3) Construct KNN for the 64-dimensional shallow features of each point. Expand the dimensionality of the point cloud (horizontal point cloud dimensionality, nonvertical feature dimensionality) with a directed graph, add the local information of K neighborhood points like the feature clustering, and then pool the most representative neighborhood information by MLP (64, 128) after mapping the point features with aggregated neighborhood information to 128 dimensions.

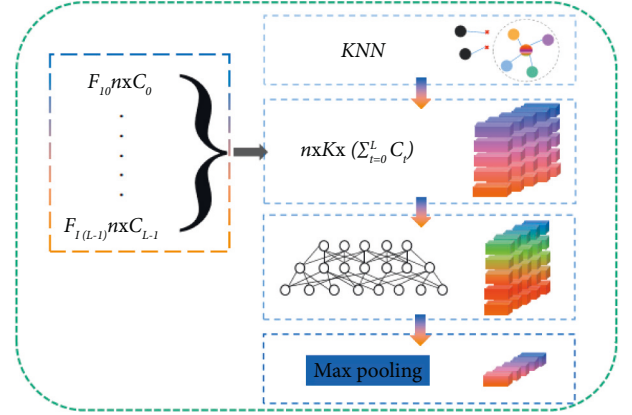


FIGURE 4: Graph feature space value filtering.

- (4) Use the idea of residual network to connect the shallow features of the point cloud with the K -dimensional pooling by using a jump connection. After the graph model generates features that consider the local information, the original point cloud information is maintained in the deep network as the abstraction of features deepens to enhance the prediction capability of the network. In contrast to the edge function selection, which only considers local features, the information of nodes and neighboring points is considered while reducing the number of parameters to improve efficiency.
- (5) The fused features are mapped to higher dimensions using MLP (1024), and fine-grained features at different scales are captured using spatial pyramid pooling to further enhance the feature abstraction capability of the network.
- (6) The high-dimensional features of N points are connected with the fused features to improve the prediction accuracy of the network. Finally, the fully connected layer is entered for feature dimensionality reduction, and set the dropout layer to prevent model overfitting, and obtain the probability matrix of N points corresponding to M categories to achieve the key point extraction of the point cloud.

4. Experiments and Results

The experimental setup along with experiments conducted is discussed in this section. The results produced are also presented.

4.1. Experimental Setup. ModelNet is a standard dataset for classification of 3D models publicly available at Princeton University, with 12715 classification models in 662 classes, divided into two classes, ModelNet10 and ModelNet40. ModelNet10 contains 4899 models in 10 classes, with 3991 training samples and 908 testing samples; ModelNet40 contains 12311 models in 40 classes, including 9843 for training and 2468 for testing. The results of testing using this class division are called instance accuracy. If the first 20

models in each category test catalog are used as the test set and the first 80 models in the training catalog are used as the training set, the test results are called category accuracy. In this experiment, the example accuracy is selected as the test result.

The experimental hardware and software environment is shown in Table 2, and the model parameters are set as shown in Table 3. The training process loss convergence is shown in Figure 5.

4.2. Experimental Results and Analysis. The comparison of accuracy and number of parameters in ModelNet40 and ModelNet10 are shown in Figures 6–8, respectively. The classification accuracy of the propose method is about 92.78% on ModelNet40 and 94.2% on ModelNet10 with 0.61 M parameters, which is the best result in the same number of parameters. It is higher than all multiview-based classification networks and most voxel and point cloud-based classification networks.

The experimental results are analyzed and discussed below:

- (1) Comparison with multiview fusion-based network: The multiview fusion-based approach uses multiple projections of different fixed views to input the rendered image into the feature space value filtering neural network, which performs single-view key point extraction from the projected rendered image, and the input is required to be a continuous model. In contrast, the IEGCNN network model in this paper takes sparse and disordered point clouds as input, and the network model is more lightweight, with only about 0.4% of the parameters based on multiview, and the classification accuracy on ModelNet40 and ModelNet10 is improved by about 2.08% and 1.4%, respectively, compared with the pairwise network, which indicates that IEGCNN can well learn the essential features of the point cloud model.
- (2) Comparison with voxel-based network: The voxel-based network can be built deeper and the network structure can be more complex due to the advantage of deep learning. The classification accuracy of the proposed method is only lower than that of VRNEnsemble, which is one of the many voxel-based methods, and the classification accuracy decreases by about 2.76% and 2.94% on the ModelNet40 and ModelNet10 datasets, respectively. VRNEnsemble trains voxel-based variable autoencoders, which are designed on the basis of ResNet, and the deep ResNet can be seen as an integration of shallow neural networks of different depths; ResNet enhances the flow of gradients by jumping connections. The voxel-based network takes full advantage of deep learning. The network is 45 layers' deep, the network architecture is complex, and as the depth increases, the network can better approximate the objective function through many nonlinear

TABLE 2: Experimental platform and configuration.

Operating systems	Linux Centos7
Operations acceleration library	RTX 2080Ti
Languages	CUDA 10. 1 + cuDNN7.5
GPU	Pytorch 1.3
Framework	Python 3. 5. 2

mappings and improved feature representations. For these reasons, the classification accuracy of VRNEnsemble-based network is higher than that of the method in the paper, but the training of this network requires encoding and decoding operations for the 3D voxel model, and the training time is the longest, which takes 6 days. At the same time, the voxel-based deep learning networks cannot be directly applied to the disordered and sparse point cloud model and require complicated voxelization operations. The designed network model contains only three feature space value filtering layers and one fully connected layer, which can quickly and directly process point clouds. The network parameters account for about 0.7% of the VRNEnsemble method. The eigenspace value filtering layer of the paper is 3 layers, while the eigenspace value filtering layer of the VRN architecture is 45 layers, which has advantages in terms of light weight and real-time.

- (3) Comparison with PointNet: The designed network architecture has about 1% of the network parameters of PointNet. The classification accuracy is improved by about 3.58% and 1.12% using ModelNet40 and ModelNet10 datasets, respectively. The network parameters of IEGCNN are reduced by about 0.19 M compared to PointNet (Vanilla), but the classification performance is improved by about 5.58% on ModelNet40 and 2.24%. The experimental results show that the present network architecture can meet the requirements of classification accuracy and lightweight in the network with the original point cloud as input.
- (4) Comparison with LDGCNN and DGCNN: The classification accuracy of IEGCNN is improved by about 0.58% compared with DGCNN, and the network parameters are about 30% of DGCNN. Although the classification accuracy on ModelNet40 is about 0.12% lower than that of LDGCNN model, the number of feature space value filtering layers is about 60% of that of LDGCNN and the training time is about 1/3 of that of LDGCNN model. Therefore, simply increasing the number of channels and the number of fully joined layers does not necessarily improve the overall performance of the network architecture.
- (5) Comparison with 3DmFV and Point2Sequences networks: 3DmFV uses Fisher vectors as the input of the feature space value filtering neural network and voxels the point cloud into a standard 3D grid, which solves the problem of disorder of the point cloud.

TABLE 3: Experimental parameter setting.

Batch volume	Data set	Number of points	K -nearest neighbor points	Optimizer	Learning rate	Number of training sessions
ModelNet40	1024	20	SGD	0.001	32	250
ShapeNetPart	2048	40	SGD	0.001	14	200
S3DIS	4096	20	SGD	0.001	14	100

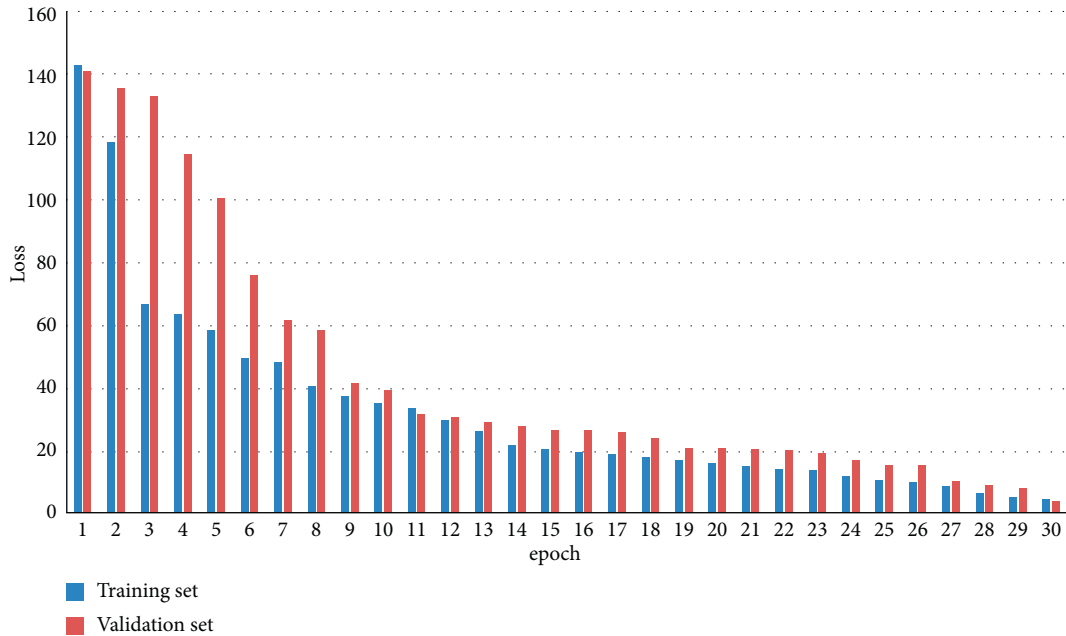


FIGURE 5: Schematic diagram of training process loss convergence.

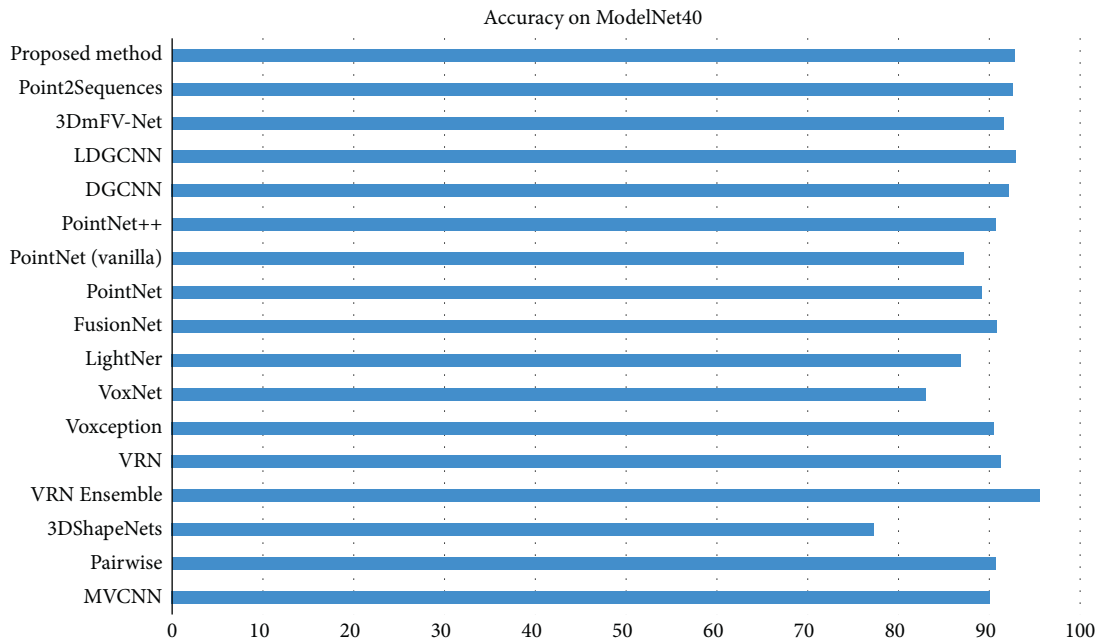


FIGURE 6: Accuracy comparison in ModelNet40.

Since Fisher vectors are computed using a voxelized grid, they are computationally intensive and memory consuming, which leads to information loss through manual key point extraction. Point2Sequences is a

recurrent neural network-based model that uses a point cloud sequence learning model to capture the correlation between different regions within a local area of the point cloud by capturing the correlation

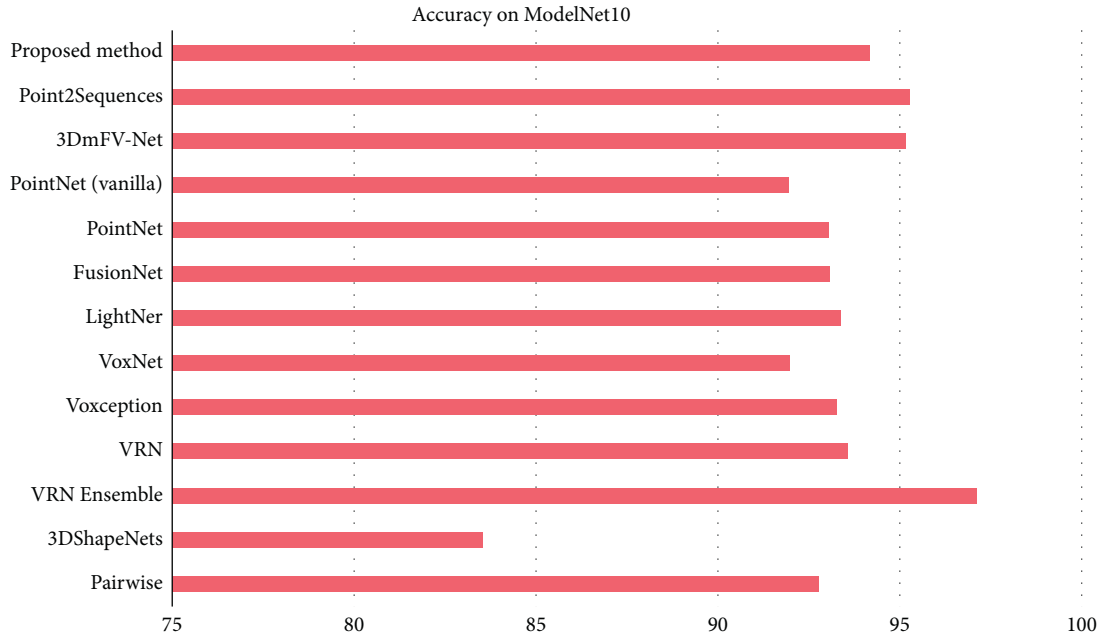


FIGURE 7: Accuracy comparison in ModelNet10.

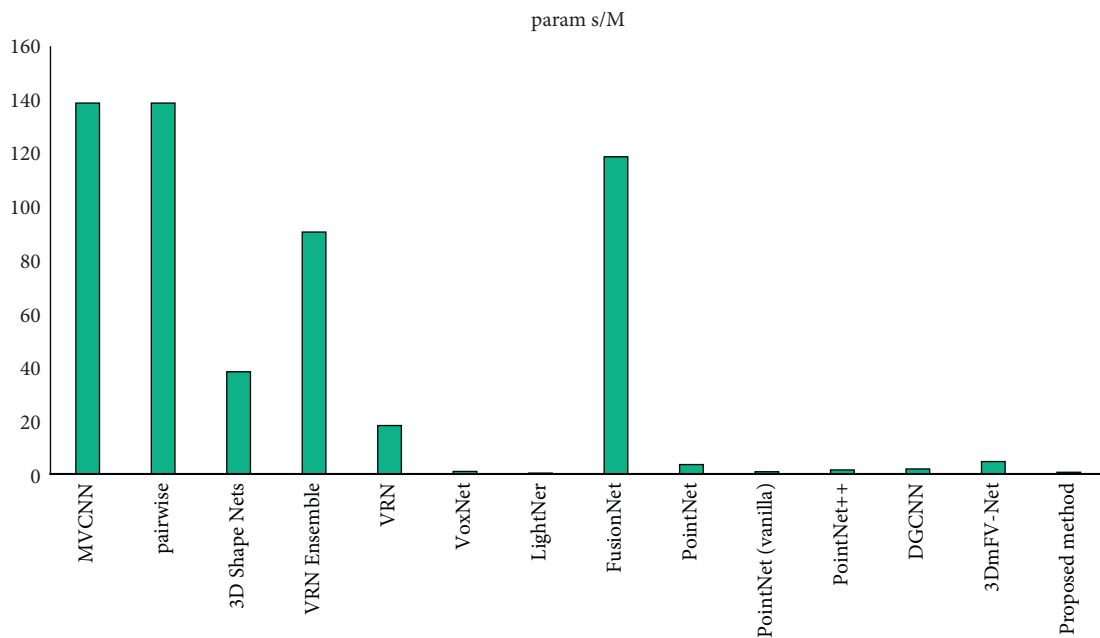


FIGURE 8: Comparison of number of parameters.

between different regions in the local area of the point cloud; the features of all the local areas are input into the coder-decoder of a recurrent neural network to achieve the aggregation of regional features. The proposed IEGCNN network improves the classification accuracy by about 1.68% over 3DmFV and 0.18% over Point2Sequences on ModelNet40 but decreases by about 1% and 1.1% over 3DmFV and Point2Sequences on ModelNet10, respectively, because the proposed architecture discards the number

of feature space value filtering layers and the number of fully connected layers, and the number of nodes is significantly reduced because the traditional feature transformation layer is dropped. In the ModelNet10 dataset, due to the distribution pattern of training and testing samples and the limitation of the number of models, the features of the point cloud model are not fully extracted.

- (6) Analysis of the generalizability of the streamlined model: The framework of the network model is

designed based on PointNet, and considering the problem of insufficient extraction of local key points in PointNet and PointNet++ network structures, a method of introducing the directional information of points is proposed combining edge index jump links, reducing the size of the network, reducing the number of feature space value screening layers in the model, and gradually changing the number of channels in each feature. The optimal model parameters are determined by gradually changing the number of channels in each spatial value filtering layer. Theoretically, the reduction of the network structure will cause a certain decrease in the classification accuracy. To compensate for the reduced accuracy caused by the reduction of the eigenspace value filtering network, the number of channels in the second eigenspace value filtering layer (Block2) was increased from 64 to 128 in the original model, and the key point extraction process was optimized by using indexed feature transfer to reduce feature loss and make the extraction more comprehensive. From the analysis of the experimental data, the streamlined network model can quickly process the whole point cloud model and improve the classification accuracy while reducing the number of parameters, which has strong universality.

5. Conclusion

With the continuous development of deep learning, the technology of 3D point cloud target recognition is more and more applied to the fields of unmanned driving and intelligent robotics. The unmanned vehicles and intelligent robots can be widely used in logistics sorting and transportation, and the data collected by unmanned vehicles and intelligent robots are mostly in the form of point clouds. Among them, the classification and segmentation of point clouds is the basis of 3D point cloud target recognition, and more and more research works are carried out around point cloud classification and segmentation.

A hierarchical key point extraction framework is proposed to solve the problem of modeling the local geometric structure between points. By analyzing point cloud models such as PointNet, PointNet++, and DGCNN and their features in local key point extraction, the indexed edge geometric feature spatial value screening neural network IEGCNN is proposed, which extracts features from each point and its neighborhood, calculates the distance between the center point and the points within its neighborhood, and adds the point orientation information to the edge feature spatial value screening network. The relationship between points in the edge network architecture is projected onto a 3D coordinate system and decomposed into three orthogonal bases, and the geometric structure between two points is modeled by feature aggregation based on the angle between the edge vector and the base vector and the distance between the center point and the neighboring points. It has the capability of fast processing of point cloud data by significantly reducing the training and recognition time.

This work not only achieves better results in classification tasks, but also provides an idea to solve the problem of real-time target detection network, which has a broad application prospect in the deployment of removable devices and real-time processing.

Data Availability

The datasets used during the current study can be obtained from the author upon reasonable request.

Conflicts of Interest

The author declares that there are no conflicts of interest.

References

- [1] S. Ding, S. Qu, Y. Xi, and S. Wan, "A long video caption generation algorithm for big video data retrieval," *Future Generation Computer Systems*, vol. 93, pp. 583–595, 2019.
- [2] Z. Gao, Y. Li, and S. Wan, "Exploring deep learning for view-based 3D model retrieval," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 16, no. 1, pp. 1–21, 2020.
- [3] Z. Gao, H. Xue, and S. Wan, "Multiple Discrimination and Pairwise CNN for view-based 3D object retrieval," *Neural Networks*, vol. 125, pp. 290–302, 2020.
- [4] S. Ding, S. Qu, Y. Xi, and S. Wan, "Stimulus-driven and concept-driven analysis for image caption generation," *Neurocomputing*, vol. 398, pp. 520–530, 2020.
- [5] S. Wang, G. Cai, and M. Cheng, "Robust 3D reconstruction of building surfaces from point clouds based on structural and closed constraints," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 170, pp. 29–44, 2020.
- [6] M. Zhao, C. Chen, L. Liu, D. Lan, and S. Wan, "Orbital collaborative learning in 6G space-air-ground integrated networks," *Neurocomputing*, vol. 497, pp. 94–109, 2022.
- [7] C. Chen, Y. Zeng, H. Li, Y. Liu, and S. Wan, "A Multi-Hop Task Offloading Decision Model in MEC-Enabled Internet of Vehicles [J]," *IEEE Internet of Things Journal*, 2022.
- [8] C. Chen, J. Jiang, Y. Zhou, N. Lv, X. Liang, and S. Wan, "An edge intelligence empowered flooding process prediction using Internet of things in smart city," *Journal of Parallel and Distributed Computing*, vol. 165, pp. 66–78, 2022.
- [9] R. A. Brown and G. B. Pasternack, "Hydrologic and topographic variability modulate channel change in mountain rivers," *Journal of Hydrology*, vol. 510, pp. 551–564, 2014.
- [10] C. Kang, Y. K. Lai, Y. X. Wu, R. Martin, and S. M. Hu, "Automatic semantic modeling of indoor scenes from low-quality RGB-D data using contextual information [J]," *ACM Transactions on Graphics*, vol. 33, no. 6, pp. 1–12, 2014.
- [11] H. Zhang, Z. Liang, C. Li et al., "A practical robotic grasping method by using 6D pose estimation with protective correction," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 99, p. 1, 2021.
- [12] A. A. Mustafa, L. G. Shapiro, and M. A. Ganter, "3D object identification with color and curvature signatures," *Pattern Recognition*, vol. 32, no. 3, pp. 339–355, 1999.
- [13] K. N. Musselman, S. A. Margulis, and N. P. Molotch, "Estimation of solar direct beam transmittance of conifer canopies from airborne LiDAR," *Remote Sensing of Environment*, vol. 136, pp. 402–415, 2013.

- [14] A. Nayegandhi, J. C. Brock, and C. W. Wright, "Small-footprint, waveform-resolving lidar estimation of submerged and sub-canopy topography in coastal environments," *International Journal of Remote Sensing*, vol. 30, no. 4, pp. 861–878, 2009.
- [15] Y. Wei, Z. Ding, H. Huang, C. Yan, J. Huang, and J. Leng, "A non-contact measurement method of ship block using image-based 3D reconstruction technology," *Ocean Engineering*, vol. 178, no. APR.15, pp. 463–475, 2019.
- [16] M. Zeybek, "Extraction of road lane markings from mobile LiDAR data [J]," *Transportation Research Record*, vol. 2675, no. 5, pp. 30–47, 2021.
- [17] H. Guan, J. Li, Y. Yu, M. Chapman, and C. Wang, "Automated road information extraction from mobile laser scanning data [J]," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 194–205, 2015.
- [18] Q. Zhang, J. Cheng, S. Wang, C. Xu, and X. Gao, "Point-selection and multi-level-point-feature fusion-based 3D point cloud classification," *Electronics Letters*, vol. 56, no. 6, pp. 290–293, 2020.
- [19] L. Yang, Y. Sheng, and B. Wang, "3D reconstruction of building facade with fused data of terrestrial LiDAR data and optical image," *Optik*, vol. 127, no. 4, pp. 2165–2168, 2016.
- [20] S. Chen, D. Tian, C. Feng, A. Vetro, and J. Kovačević, "Fast resampling of 3D point clouds via graphs [J]," *IEEE Transactions on Signal Processing*, vol. 66, no. 99, p. 1, 2017.
- [21] E. Kalogerakis, D. Nowrouzezahrai, P. Simari, and K. Singh, "Extracting lines of curvature from noisy point clouds," *Computer-Aided Design*, vol. 41, no. 4, pp. 282–292, 2009.
- [22] H. Wu, C. Liu, Y. Zhang, W. Sun, and W. Li, "Building a water feature extraction model by integrating aerial image and lidar point clouds," *International Journal of Remote Sensing*, vol. 34, no. 21, pp. 7691–7705, 2013.
- [23] Y. Ke, S. Fan, W. Zhu, A. Li, F. Liu, and X. Shi, "Feature-based reverse modeling strategies," *Computer-Aided Design*, vol. 38, no. 5, pp. 485–506, 2006.
- [24] M. Panchetti, J. P. Pernot, and P. Véron, "Towards recovery of complex shapes in meshes using digital images for reverse engineering applications," *Computer-Aided Design*, vol. 42, no. 8, pp. 693–707, 2010.
- [25] S. Sharifzadeh, I. Biro, N. Lohse, and P. Kinnell, "Abnormality detection strategies for surface inspection using robot mounted laser scanners," *Mechatronics*, vol. 51, pp. 59–74, 2018.
- [26] K. Demarsin, D. Vanderstraeten, T. Volodine, and D. Roose, "Detection of closed sharp edges in point clouds using normal estimation and graph theory," *Computer-Aided Design*, vol. 39, no. 4, pp. 276–283, 2007.
- [27] A. Frome, D. Huber, R. Kolluri, T. Bülow, and J. Malik, "Recognizing objects in range data using regional point descriptors [C]," in *Proceedings of the European Conference on Computer Vision*, pp. 224–237, Springer, Berlin, Heidelberg, 2004.
- [28] Y. Guo, F. A. Sohel, M. Bennamoun, J. Wan, and M. R. P. S. Lu, "A local feature descriptor for 3D rigid objects based on rotational projection statistics [C]," in *Proceedings of the 1st IEEE International Conference on Communications, Signal Processing, and Their Applications (ICCSPA)*, pp. 1–6, Sharjah, United Arab Emirates, February 2013.
- [29] F. Tombari, S. Salti, and L. Stefano, "Unique signatures of histograms for local surface description [C]," in *Proceedings of the Euro Conference on Computer Vision*, Berlin, 2010.
- [30] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz, "Aligning point cloud views using persistent feature histograms [C]," in *Proceedings of the 21st IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nice, France, September 2008.
- [31] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration [C]," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3212–3217, Kobe, Japan, August 2009.
- [32] C. R. Qi, H. Su, K. Mo, and L. J. P. Guibas, "Deep learning on point sets for 3D classification and segmentation [C]," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 652–660, Honolulu, HI, USA, 2017.
- [33] Z. Gojcic, C. Zhou, J. D. Wegner, and A. Wieser, "The perfect match: 3d point cloud matching with smoothed densities [C]," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5545–5554, Long Beach, CA, USA, 2019.
- [34] X. Huang, Z. Jian, L. Fan, Q. Wu, and C. Yuan, "A systematic approach for cross-source point cloud registration by preserving macro and micro structures [J]," *IEEE Transactions on Image Processing*, vol. 26, no. 7, p. 1, 2017.
- [35] J. S. Yoon, M. Sagong, J. S. Lee, and Ks Lee, "Feature extraction of a concrete tunnel liner from 3D laser scanning data," *NDT & E International*, vol. 42, no. 2, pp. 97–105, 2009.
- [36] I. F. Leong, J. J. Fang, and M. J. Tsai, "Automatic body feature extraction from a marker-less scanned human body," *Computer-Aided Design*, vol. 39, no. 7, pp. 568–582, 2007.
- [37] R. Y. Takimoto, M. d S. G. Tsuzuki, R. Vogelaar et al., "3D reconstruction and multiple point cloud registration using a low precision RGB-D sensor," *Mechatronics*, vol. 35, no. January, pp. 11–22, 2016.
- [38] R. Ali, A. M. Khatak, F. Chow, and S. Lee, "A case-based meta-learning and reasoning framework for classifiers selection," in *Proceedings of the 12th international conference on ubiquitous information management and communication*, pp. 1–6, Langkawi, Malaysia, 2018 January.
- [39] R. Ali, S. Lee, and T. C. Chung, "Accurate multi-criteria decision making methodology for recommending machine learning algorithm," *Expert Systems with Applications*, vol. 71, pp. 257–278, 2017 Apr 1.