Engineering Applications of Artificial Intelligence 120 (2023) 105956

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Semi-supervised active learning hypothesis verification for improved geometric expression in three-dimensional object recognition

Zhenhao Wang [a], Rui Xu [b,*], Tingyuan Nie [a], Dong Xu [c]

[a] School of Information and Control Engineering, Qingdao University of Technology, No. 777 Jialingjiang East Road, West Coast new area, Qingdao, 266520, Shandong, China
[b] Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun, 133033, Jilin, China
[c] Department of Electrical Engineering and Computer Science, Bond Life Sciences Center, University of Missouri, Columbia, MO 65211-2060, USA

ARTICLE INFO

Keywords:
K-means++
Hypothesis verification
3D object recognition
Geometric expression
FPFH descriptor extraction

ABSTRACT

Efficient three-dimensional (3D) object recognition plays an important role in the 3D reconstruction of light-field displays. However, presently, the error rate of 3D implicit shape object recognition remains high, because the local features are sparse in the geometric expression of 3D reconstruction. To address this issue, a hypothesis verification method based on semi-supervised active learning-based K-means++ combined with 3D feature extraction is proposed. The proposed approach consists of the offline and online phases. The algorithm time complexity is $\mathcal{O}(n)$ and $\mathcal{O}(n^2)$, respectively. The offline phase includes keypoint detection, normal estimation, fast point feature histograms (FPFH) descriptor extraction, geometric word weight saving, and indexing structure construction. In addition to the FPFH extraction, the online phase includes nearest geometric word searching, corresponding direction and center voting, and non-maximum suppression. Comparative experiments were conducted in which the models and scenes were tested on the 3D datasets Mian and Tosca that is high-resolution. The experimental results demonstrate that the proposed method resolves the low recognition rate problem of 3D implicit objects, with the highest 3D intersection over union (IoU) reaching 88.89%.

1. Introduction

The glasses-free tabletop three-dimensional (3D) display is an attractive technology that allows multiple individuals around a table to view reconstructed 3D objects simultaneously (Ren et al., 2020). The light-field display, which uses a geometric optical directional screen and projection technology, has progressed significantly because of advances in high-definition pixels, high-end graphics processing units (GPUs), and image processing technology.

The light-field display is a high-quality and high-resolution color dynamic 3D display compared to the traditional parallax 3D display technology, and it can also display complex texture and illumination shadows. However, a fundamental trade-off with the achieved spatial resolution remains for near-eye light-field displays that require portability. The computational power must be increased for the rapid synthesis of a high-quality light field (Gao et al., 2021).

To date, the light-field display has obtained real-time updates and dynamic processing of light-field 3D reconstruction (Wang et al., 2022), including various enhancement schemes, data compression algorithms, and parallel operations of GPUs. Moreover, the geometric expression of 3D reconstruction requires extensive calculation; for example, point cloud annotation, particularly on the pixel level, requires more time and energy than image annotation (Chen et al., 2019; Bletterer et al., 2020; Tang et al., 2022).

Multi-fringe projection profilometry can achieve high accuracy and robustness in the 3D reconstruction of static objects. An effective method based on an automated transmission line for reconstructing the 3D shapes of rigid moving objects was proposed in Wang et al. (2020b). Light-field 3D reconstruction can recognize and track 3D object models in real-time, thereby providing a realistic virtual environment because of its unique advantages in environmental awareness and situation assessment.

Existing methods for 3D object recognition (Cho and Kang, 2021) can be roughly classified into the following three types: (i) two-dimensional (2D) feature extraction (mapping from 2D to 3D) (Zhang et al., 2020; Kim et al., 2021; Peng et al., 2021); (ii) machine learning (Feng et al., 2020; Liu et al., 2021); and (iii) 3D feature extraction (Zhu et al., 2021; Song et al., 2022). However, 3D object recognition based on 2D feature extraction can only capture the image

* Corresponding author.
E-mail addresses: wangzhenhao@qut.edu.cn (Z. Wang), xur@ciomp.ac.cn (R. Xu), tynie@qut.edu.cn (T. Nie), xudong@missouri.edu (D. Xu).

https://doi.org/10.1016/j.engappai.2023.105956
Received 18 August 2022; Received in revised form 9 January 2023; Accepted 2 February 2023
Available online 10 February 2023
0952-1976/

features of the light-field viewpoint in the specific direction of the light field.

Machine learning can solve several problems in 2D feature extraction methods by reconstructing 3D objects from 2D images, in which a category definition, prior knowledge, and assumptions are required (Fu et al., 2021). Machine learning may be supervised or unsupervised. The former approach requires numerous labeled training data, which are generally difficult to obtain in the processing of light-field 3D reconstruction, whereas the latter cannot achieve the same performance as the former.

Active learning is a type of machine learning in which the most valuable samples are actively selected for labeling. Its purpose is to use as few high-quality samples as possible to achieve the best performance of the model. The active learning method can improve the gain of the samples and labeling. On the premise of a limited labeling budget, maximizing the model performance can improve the data efficiency from the perspective of samples. Moreover, semi-supervised learning can achieve excellent performance when insufficient labels are available. Therefore, a semi-supervised active learning method is proposed in this study.

The multiple knowledge representation (Yang et al., 2021; Pan, 2021) is a general framework to improve the feature quality via structural knowledge integration, which is to integrate structural features as extra knowledge into the learning. 3D features exhibit distinct advantages, including global and local features. These features contain angular and spatial resolution and light-field geometric expressions. Various 3D feature extraction methods are available such as the multi-directional affine registration method based on the shape features and statistical characteristics of point cloud (Wang et al., 2018).

Another registered point cloud method is based on the Cauchy mixture model in rigid registration (Wang et al., 2020a). In Wang et al. (2019), a strategy for 3D object recognition in the 3D reconstruction of the light-field display was presented. This method is based on probability and a mathematical statistics algorithm, which simplifies the 3D object recognition, increases the efficiency, and reduces the computational consumption for light-field 3D reconstruction. However, several problems appear such as the low recognition rate of implicit shapes and slow online recognition speed.

Therefore, this study focuses on the low recognition rate of implicit shapes. The large interval principle is adopted to reduce the dimensionality and to improve the low dimensional local feature recognition. A semi-supervised active learning method combined with 3D feature extraction is proposed to construct the nonlinear manifold structure of 3D objects. In this manner, the similarity between 3D features and 3D objects can be calculated accurately. Furthermore, the 3D object centroid is used as the recognition result to improve the online recognition speed. Consequently, the proposed method can improve the accuracy and efficiency of the geometric consistency of the light-field display.

The remainder of this paper is organized as follows: Section 2 presents the conceptual and theoretical framework of the proposed approach. Section 3 outlines the experimental results. Section 4 are summarized the conclusions and future directions. Finally, the discussion is in Section 5.

2. Proposed approach

2.1. Overview

A block diagram of the overall light-field acquisition-display data-processing based on the proposed method is depicted in Fig. 1. The brown solid line box represents the image processing stage of 3D reconstruction. The red densely dashed box indicates the offline and online parts. The purple part of the diagram represents the core of the proposed method. The 3D object recognition process consists of offline and online parts. Algorithms 1 and 2 present the detailed steps; the algorithm time complexity is $\mathcal{O}(n)$ and $\mathcal{O}(n^2)$, respectively. Whether supervised or unsupervised machine learning is used, the offline and online parts are the sequential and successive phases of the machine learning. Moreover, the exact offline part is the adaptive basis for the online part for real-time 3D object recognition.

Algorithm 1: Offline part of the proposed method

```
Data: Model files train_clouds
Result: Save models to a training model file_train_model
Initialization;
while i_cloud < number_of_training_clouds − 1 do
    //Load the color and depth images of the 3D model.
    LoadPCDFile ∗ file_list_train_model_cloudl[i_cloud ∗ 2 + 1];
    //Save the point cloud and parameter vector for the training.
    training_clouds.push_back(train_clouds);
    training_normals.push_back(train_normals);
    training_classes.push_back(train_classes);
    //Extract FPFH feature descriptors of models.
    FPFHEstimation feature_estimator;
    //Set K-means++ method parameters.
    setFeatureEstimator(feature_estimator);
    setTrainingClouds(training_clouds);
    setTrainingNormals(training_normals);
    setTrainingClasses(training_classes);
    setSamplingSize(2.0f);
    if i_cloud ≥ number_of_training_clouds − 1 then
        //Save models to a training model file.
        saveModelToFile(file_train_model);
        //Training models.
        train(train_clouds);
    else
        go back to the beginning of training models;
    end
end
```

In the offline part, first, the color, depth, and point cloud information of the 3D model is obtained. Second, keypoint detection and normal estimation of the point cloud are performed. As the point cloud that is obtained by the RGB-D camera is discrete and sparse and the number of 3D coordinate points of the sparse point cloud is relatively small, it is necessary to apply normal estimation processing to expand the sparse point cloud into a dense point cloud to enrich the 3D information of the model. Third, the fast point feature histogram (FPFH) descriptors of the model are extracted as the data for the semi-supervised active learning based on K-means++. Fourth, labeling is used to select the K centroids, which manually specify the initial values in a specific order, and each point of the FPFH feature is assigned to the nearest centroid. Fifth, by computing the minimum distance between the keypoint and centroid, the keypoint is reassigned to the nearest centroid. Finally, following geometric consistency grouping, the keypoint is no longer reassigned, and the iterations end, at which point the geometric word weight is saved to construct an index structure.

In the online part, first an RGB-D camera is used to capture the color and depth image, following which the point cloud information of the 3D model and scene is obtained. Second, keypoint detection and normal estimation of the point cloud are obtained. As the FPFH descriptors are based on the relationship between the point and its neighbor with normal estimation, the FPFH estimation is performed after the normal estimation. Moreover, the 3D reconstruction involves filling the hollow part of the dense point cloud in the 3D reconstruction phase. Thus, in the method proposed in Wang et al. (2019), keypoint sampling and Monte Carlo random sampling of the model and scene are performed. Thereafter, the signature of histograms of orientations (SHOT) description of the model and scene is extracted, and K-dimensional (KD) -tree searching is used to index the spatial point cloud information. In this manner, geometric consistency correspondence grouping estimation between the model and scene is achieved.

Because of the low recognition rate of 3D implicit shape objects, this study presents a semi-supervised active learning hypothesis verification method that combines FPFH estimation with the index structure result of the offline part to achieve the nearest geometric word searching. Moreover, the corresponding direction, center voting algorithm, and

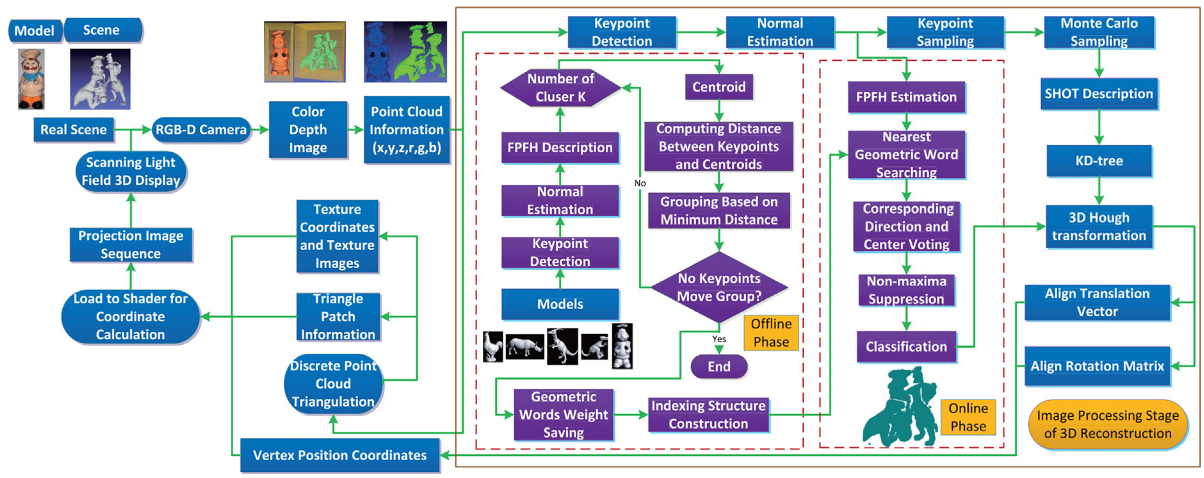

Fig. 1. Block diagram of overall light-field acquisition-display data processing based on proposed method.

Algorithm 2: Online part of the proposed method

input : Scene files *test_clouds*, training model *file_train_model*
output: Visualize the object found in the original scene with the red mark.

```
Initialization;
for i_point ← 0 to i_point < testing_cloud− > points.size() do
    for i_vote ← 0 to i_vote < strongest_peaks.size() do
        point.x = strongest_peaks[i_vote].x;
        point.y = strongest_peaks[i_vote].y;
        point.z = strongest_peaks[i_vote].z;
        vote_list− >
          findStrongestPeaks(strongest_peaks, testing_class, radius, sigma);
        if i_vote ≥ strongest_peaks.size() then
            colored_cloud− > height+ = strongest_peaks.size();
            //Add strongest peaks to colored_cloud, used for displaying the red
              centroid of found object
            colored_cloud− > points.push_back(point);
        else
            go back to the beginning of finding object;
        end
    end
    foreach i_point ≥ testing_cloud− > points.size() do findStrongestPeaks;
end
while !viewer.wasStopped() do
    viewer.spin();
    visualizationcolorh(testing_cloud);
    viewer.addPointCloud(colored_cloud);
end
```

non-maximum suppression algorithm improve the recognition rate of 3D implicit shape objects. Finally, all results of the geometric consistency are marked using the 3D Hough transform. At this point, the 3D object recognition is completed, and the precise rotation matrix, translation vector of the model and scene, and vertex position coordinates can be obtained, which are required for the texture mapping. Texture mapping involves mapping the texture information from 2D space into 3D space for light-field display. In the remainder of the light-field acquisition-display system in Fig. 1, discrete point cloud triangulation is obtained from the 3D point cloud information to determine the texture coordinates and triangle patch information. Subsequently, the two groups of coordinate information are sent to the GPUs, in which the vertex and segment shaders perform mapping, and the projection image sequence is output to the light-field display.

2.2. Keypoint detection

The keypoint detection (Prakhya et al., 2016) is used to reduce an input point cloud and to collect fewer key points from the original point cloud. In this study, in combination with using the foreground and background scene point segmentation strategy, by increasing the weight of the foreground scene, the ratio of key points is increased, thereby improving the object detection accuracy. The ultimate goal of this method is to collect n key points $M = p_1, p_2, \dots, p_n$ from the original point cloud P. The main steps are as follows:

(1) The first point p_1 is randomly selected from point cloud P; the Euclidean distance from all points to p_1 in P except p_1 is calculated, and the point with the largest distance from p_1 in P is determined and marked as p_2. p_1 and p_2 are the initial values of set M.

(2) The distance between all points and key point set M, except for p_1 and p_2 in set P, is calculated (only two points p_1 and p_2 exist in set M at this time). The distance D_{P_i-K} from any point p_i in point cloud P to key point set M is defined as the minimum distance from point p_i to all points in set M.

(3) According to the foreground and background scene point segmentation strategy, weighted processing is performed with the D_{P_i-K} of each point in set P, and the weight value is the normalized value (the range is from 0 to 1). The weight value is close to 1 for a foreground scene, whereas it is close to 0 for a background scene. However, the minimum value b and maximum value f of the weight value are not normalized. The Sigmoid function is required to scale these values to the interval $[0, 1]$ so that the value can be used as the segmentation weight. If b is larger, the point is a background point. When the weight is set to a smaller value bw, the D_{P_i-K} value of the foreground scene will be larger following weighting. Moreover, the D_{P_i-K} value of the background scene is smaller. Therefore, according to the D_{P_i-K} value of each point, if the point is the farthest point for sampling, the foreground scene point is more likely to be sampled.

(4) The point with the largest distance D_{P_i-K} value to key point set M is selected as key point p_3 in point cloud P. At this time, $M = p_1, p_2, p_3$.

(5) The distance between each remaining point in point cloud P and point set M is repeatedly calculated, and the weight is adjusted according to the semanteme segmentation strategy. When the distance D_{P_i-K} reaches the preset maximum of key point set M from the furthest point p_{max} to key points set M, all steps are completed.

2.3. FPFH descriptors

The FPFH (Rusu et al., 2009) is an improvement of the point feature histogram (PFH) (Rusu et al., 2008); the PFH feature description is coded based on the spatial geometric relationship between feature points and their neighbors, and its calculation principle is depicted in Fig. 2. The center feature point P_q of the pentagon is a query point, and k points ($P_{k1} \sim P_{k5}$) (the purple line enclosed by the light green circle) are searched in its neighborhood. For the query point and k-neighbors, $k+1$ points can be obtained by pairing in pairs $k(k+1)/2$

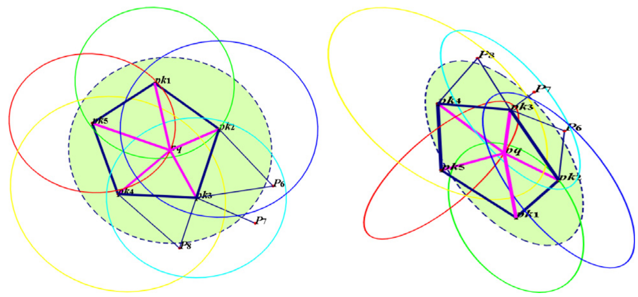

Fig. 2. Calculation principle of FPFH.

point pairs. A query point P_q is connected only to its neighborhood points ($P_{k1} \sim P_{k5}$) to estimate its simple point feature histogram (SPFH). Compared to the standard calculation of the PFH, no interconnection exists between neighborhood points ($P_{k1} \sim P_{k5}$). All points in the point cloud dataset are required to perform this calculation to determine the SPFH. Each direct neighbor is connected to its own neighborhood points ($P_{k1} \sim P_{k5}$), and the resultant SPFH is weighted together with that of query point P_q to obtain the final FPFH of the point. The directly connected points of the FPFH are represented by dark blue lines in Fig. 2, where the thicker connections contribute to the FPFH twice, whereas the others that are indirectly connected are indicated by thin dark blue lines.

The FPFH provides the weighted statistics for each SPFH in the neighborhood, the formula for which is as follows:

$$FPFH(P_q) = SPFH(P_q) + \frac{1}{k}\sum_{i=1}^{k}\frac{1}{\omega_k}\cdot SPFH(P_k) \tag{1}$$

where ω_k is the distance between query point P_q and its neighborhood points P_k.

The FPFH takes advantage of triples (α, ϕ, θ) to describe the deviation of the normal vectors between the query point and neighborhood points of the light green circle in Fig. 2. The statistics of each component in the triple are normalized to $[0, 100]$ and are divided into 11 intervals. A total of 33 angle intervals are formed for the number statistics of the triple values. Thus, a 33-dimensional eigenvector is obtained.

2.4. K-means++

The K-means algorithm, which was proposed by MacQueen in 1967 (MacQueen et al., 1967), divides each sample into clustering with the nearest center (mean). The K-means algorithm is an important method in data mining. It provides an iterative repartition strategy: the Euclidean distance between each point and the cluster center is continuously iterated and optimized until the dataset is divided into the K predefined clusters, following which the iterative process is completed.

For every dataset $X = \{x_1, x_2, \dots, x_N\}$, the K-means algorithm provides a K-partition $\{X_l\}_{l=1}^{k}$. Therefore, if $\{C_1, C_2, \dots, C_K\}$ is used to represent the centers of the K-partitions, the objective function is expressed as follows:

$$E = \sum_{l=1}^{K}\sum_{x_i \in X_l}\|x_i - C_l\|^2 \tag{2}$$

The basic K-means algorithm consists of the following three steps:

(1) All samples are divided into K initial clusters.

(2) A sample from the sample-set is categorized into the cluster when the Euclidean distance is the nearest to the center of this sample. In this case, the center of the sample is the standardized or non-standardized mean of the data.

(3) The center of the two clusters is recalculated, until all samples can no longer be redistributed.

The final clustering result relies on the selection of the initial group or, to a certain extent, the initial center. Experimental results have demonstrated that the most important changes occur in the first redistribution.

Because the proposal of the K-means algorithm, it has been used extensively in many fields such as computer vision, data mining, and gene discovery. However, several problems occur in K-means clustering. For example, the K-mean is highly dependent on the selection of the initial points, which may affect the clustering efficiency and performance. Therefore, this study presents a semi-supervised active learning method based on K-means++, which manually defines the order of the initial values by tagging. K-means++ has been experimentally demonstrated to be better than the K-means method in terms of convergence and running time, which improves the robustness of the algorithm. However, as K-means++ may select outliers or low-density points as the center points, the clustering results may not be ideal.

The K-means++ algorithm is based on the K-means algorithm and improves the selection method of the initial clustering center. When the nth initial clustering center is selected ($n \in [2, k]$, k is the number of clusters), the further points from the $(n-1)$ ahead clustering center have a higher probability of being selected. The basic principle is as follows (Arthur and Vassilvitskii, 2006):

Assuming that a dataset contains N samples $\alpha = \{x_i | x_i = (x_{i1}, x_{i2}, \dots, x_{id}), i = 1, 2, \dots, N\}$, k points of set α are randomly selected as the initial clustering center $C = \{c_j | c_j = (c_{j1}, c_{j2}, \dots, c_{jd}), j = 1, 2, \dots, k\}$. The Euclidean distance d between each example point x_i in dataset α and each centroid c_j in m-dimensional space is calculated according to the following formula:

$$d(x_i, c_j) = \sqrt{\sum_{k=1}^{m}(x_{ik} - c_{jk})^2} \tag{3}$$

The formula for the probability P of each sample point being selected as the next clustering center is as follows:

$$P(i) = \frac{d^2(x_i)}{\sum(d^2(x_i))} \tag{4}$$

Using the roulette method, the next clustering center is selected, and the process is repeated and looped until K clustering centers are selected. On this basis, the Euclidean distance d between each sample point x_i and each clustering center of the sample dataset is calculated. The clustering center is determined according to the nearest distance criterion. The mean value of all sample points in each cluster is regarded as the clustering center. The clustering center is constantly updated until it does not change or the sum of square errors becomes minimal. The formula for the sum of squares of the intra-clustering errors I_{SSE} is as follows:

$$I_{SSE} = \sum_{j=1}^{k}\sum_{x_i \in \varphi_j} d(x_i, c_j) \tag{5}$$

2.5. Non-maximum suppression

Non-maximum suppression is a point convergence algorithm (Dalal, 2006; Comaniciu, 2003). In this study, non-maximum suppression is applied to the clustering problem to achieve classification improved results. It does not require prior information, such as the number of clusters, and enables each feature point to converge to several cores with the largest density. Thus, all points that converge to the same core belong to the same class. The steps are as follows:

(1) The overall spatial function $f(X)$ is constructed, which represents the density of every point X in neighborhood H. $f(X)$ must satisfy many characteristics; for example, the gradient $\nabla f(X)$ can be obtained; $\nabla f(X) = 0$ can be reduced to the iterative formula $X = \varphi(X)$, and

the iterative function $\varphi(X)$ should have H as a constant. Moreover, the Gauss kernel function must satisfy the following conditions:

$$f(X) = \frac{1}{n(2\pi)^{\frac{3}{2}}} \sum_{i=1}^{n} |H_i|^{-\frac{1}{2}} e^{-\frac{D^2[X,X_i,H_i]}{2}} \tag{6}$$

where $D^2[X, X_i, H_i] = (X - X_i)^T H(X - X_i)$ is the Mahalanobis distance between X and X_i; n is the number of clustered points (feature points); the diagonal matrix $H_i = diag(\delta_x, \delta_y)$ is known as the uncertainty matrix, which is used to adjust the number of clusters; and δ_x, δ_y denotes the size of neighborhood H_i on the x- and y-axes, respectively. A smaller neighborhood H_i indicates that more classes are clustered.

(2) All local maximum points (C_1, C_2, C_K) of $f(X)$ are determined so as to calculate the point of gradient $\nabla f(X) = 0$. In this case, k is the number of categories, and $(C_1, C_2, \dots, C_K)$ denotes the core, which characterizes the density local maximum density, as follows:

$$\nabla f(X) = \frac{1}{n(2\pi)^{\frac{3}{2}}} \sum_{i=1}^{n} |H_i|^{-\frac{1}{2}} H_i^{-1}(X_i - X) e^{-\frac{D^2[X,X_i,H_i]}{2}} \tag{7}$$

As the solution to the above equation (C_1, C_2, C_K) cannot be obtained analytically, (C_1, C_2, C_K) is determined using an iterative method. Therefore, the iterative formula is deduced as $X = \varphi(X)$ from $\nabla f(X) = 0$.

(3) Let $\nabla f(X)/f(X) = 0$; thus,

$$\widetilde{\omega}_i(X) = \frac{|H_i|^{-\frac{1}{2}} e^{-\frac{D^2[X,X_i,H_i]}{2}}}{\sum_{i=1}^{n} |H_i|^{-\frac{1}{2}} e^{-\frac{D^2[X,X_i,H_i]}{2}}} \tag{8}$$

Subsequently, $\nabla f(X)/f(X) = 0$ can be reduced to

$$\frac{\nabla f(X)}{f(X)} = \sum_{i=1}^{n} \widetilde{\omega}_i(X) H_i^{-1} X_i - (\sum_{i=1}^{n} \widetilde{\omega}_i(X) H_i^{-1}) X \tag{9}$$

That is,

$$X = \frac{\sum_{i=1}^{n} \widetilde{\omega}_i(X) H_i^{-1} X_i}{\sum_{i=1}^{n} \widetilde{\omega}_i(X) H_i^{-1}} \tag{10}$$

The above equation is the iterative formula $X_{k+1} = \varphi(X_k)$. Because of its convergence, the initial value of the iteration can be selected as any characteristic point X_i, and it must converge to one of the solutions (C_1, C_2, C_K) of $\nabla f(X)/f(X) = 0$. The iteration does not stop until X_k no longer changes.

(4) If X_i converges to C_j, X_i can cluster to class j, and all feature points that converge to C_j belong to this class. The above process is repeated for all feature points to complete the clustering of all points.

Remark. Eqs. (3), (4), and (5) imply that the training initialization (offline part) of the K-means++ is to train and save the training model, on behalf of the Euclidean distance in m-dimensional space, the probability, and the sum of squares of the intra-clustering errors. Eqs. (6)– (10) implies that classification (online part) of the K-means++ is to classify, to find the strongest peaks. The strongest peaks represent the recognition result with the red centroid.

3. Experiments

Experiments were conducted using traditional pattern recognition (Wang et al., 2019), including normal estimation, uniform keypoint sampling, Monte Carlo random sampling, SHOT descriptor extraction, KD-tree matching, and geometric consistency clustering estimation. All scenarios corresponding to each model were tested 1582 times in three databases. Fig. 3 presents part of the experimental results of the 3D object recognition using the traditional method for each model and scene in the Clutter dataset (Glover and Popovic, 2013). The experimental results indicate that the efficiency (EFF) of the traditional method exhibited an average improvement of 9.26%, with the same rate of the correct recognition (RCR). However, the recognition rate of the 3D implicit shape objects was not high in the experiments, as shown in the Banana model in the second row and the fifth column in Fig. 3. Additionally, the TikiCup model in the eighth row and the eighth column exhibits the same effect. The efficiency and the recognition rate were calculated as follows:

$$EFF = \frac{StandardTotalWorkingHours}{ActualTotalWorkingHours} \times 100\% \tag{11}$$

where the Standard Total Working Hours denoted the time involved in the traditional method and the Actual Total Working Hours denoted the time involved in the proposed method.

$$\begin{aligned} RCR &= IndividualRecognitionRate \times Weight \times 100\% \\ &= \frac{TotalNumberofRecognitionObjects}{TotalNumberofModels} \times \\ &\quad \frac{TotalNumberofModels}{TotalNumberPerClassModels} \times 100\% \\ &= \frac{TotalNumberofRecognitionObjects}{TotalNumberPerClassModels} \times 100\% \end{aligned} \tag{12}$$

In addition, performance evaluation metrics also included the 3D intersection over union (IoU) and algorithm time complexity. The 3D IoU was calculated as follows:

$$3DIoU = \frac{VolumeofOverlap}{VolumeofUnion} \tag{13}$$

where the Volume of Overlap denoted the "intersection" of two 3D objects and the Volume of Union denoted the "union" of two 3D objects. The maximum value of 3D IoU was 1, which meant the actual spatial area of the 3D object was completely coincident with the inferred spatial area. The minimum value of 3D IoU was 0, which meant there was no overlap between the actual spatial area and the inferred spatial area of the 3D object. When 3D IoU was greater than 0.5, it obtained a "good" prediction result. In experiments, the 3D IoU was equivalent to the RCR.

Complexity analysis referred to how many times the algorithm required multiplication and addition, such as the fast Fourier transform of an $N * N$ matrix, whose complexity was $\mathcal{O}\left(N^2 log N\right)$. The algorithm time complexity was expressed in Big $\mathcal{O}$ notation, defined as $T[n] = \mathcal{O}(f(n))$. The function $T[n]$ was bounded by $f(n)$. If the scale of a problem is n, the time required for an algorithm to solve the problem was $T[n]$. $T[n]$ was called the "time complexity" of this algorithm. When the input n gradually increased, the limit case of time complexity was called the "asymptotic time complexity" of the algorithm. The Big $\mathcal{O}$ notation meant that there was an upper bound of the algorithm time complexity. If $f(n) = \mathcal{O}(n)$, it was clearly true that $f(n) = \mathcal{O}\left(n^2\right)$. It gave an upper bound, but it was not a supremum. The common algorithm time complexity from small to large was: $\mathcal{O}(1) < \mathcal{O}\left(\log_2 n\right) < \mathcal{O}(n) < \mathcal{O}\left(n \log_2 n\right) < \mathcal{O}\left(n^2\right) < \mathcal{O}\left(n^3\right) < \mathcal{O}\left(n^k\right) < \mathcal{O}(2^n)$. As the problem size n increased, the above time complexity increased, and the EFF decreased.

Experiments with the proposed method were conducted using the Visual Studio 2013 platform and Point Cloud Library (PCL) 1.8.0 to improve the recognition rate of 3D implicit shape objects. The programming language was C++. The experimental databases included the Mian dataset (Mian et al., 2006) and Tosca high-resolution 3D dataset (Bronstein et al., 2008). The point cloud data (PCD) ASCII file format was used in the light-field imaging experiment. Because of the huge amount of data, it would be almost impossible to label each sample. Therefore, active learning was used to select the most valuable samples, which were subsequently manually labeled to continue with model training, improve the model performance, and improve the stability and security as far as possible. As the proposed method consists of semi-supervised active learning combined with the K-means++ method, it was not necessary to configure and adjust the parameters, such as the threshold in the C++ project based on PCL, to improve the adaptability of the 3D object recognition.

The first experiment was performed on the Mian dataset from Mian et al. (2006). Representative sampling was used to select some of the

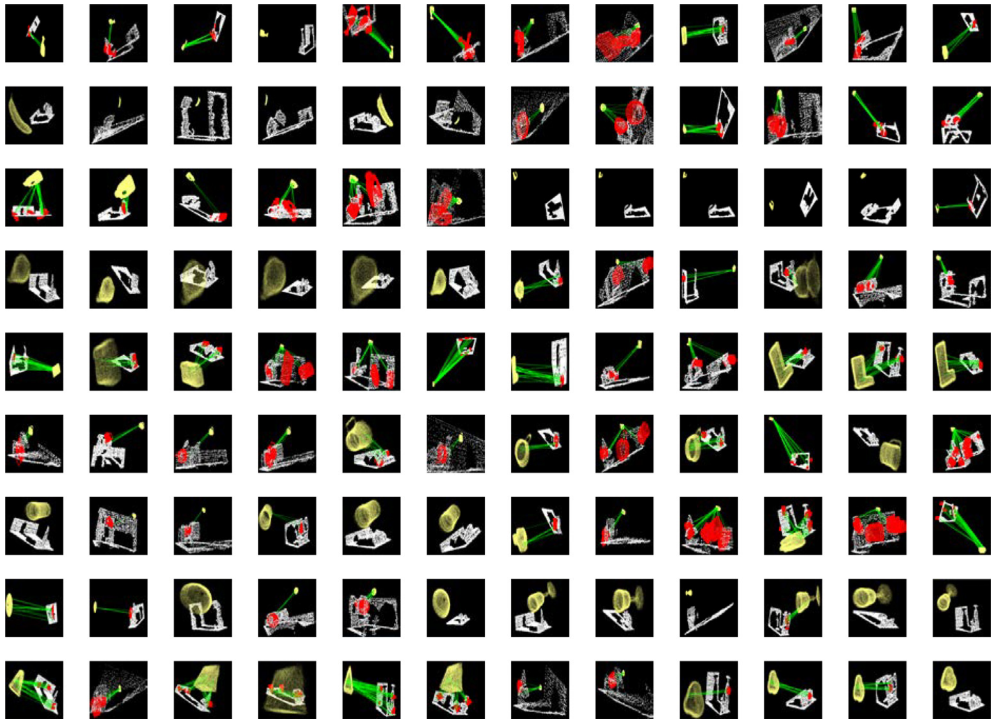

Fig. 3. Experimental results of Clutter dataset using method of traditional pattern recognition (Wang et al., 2019).

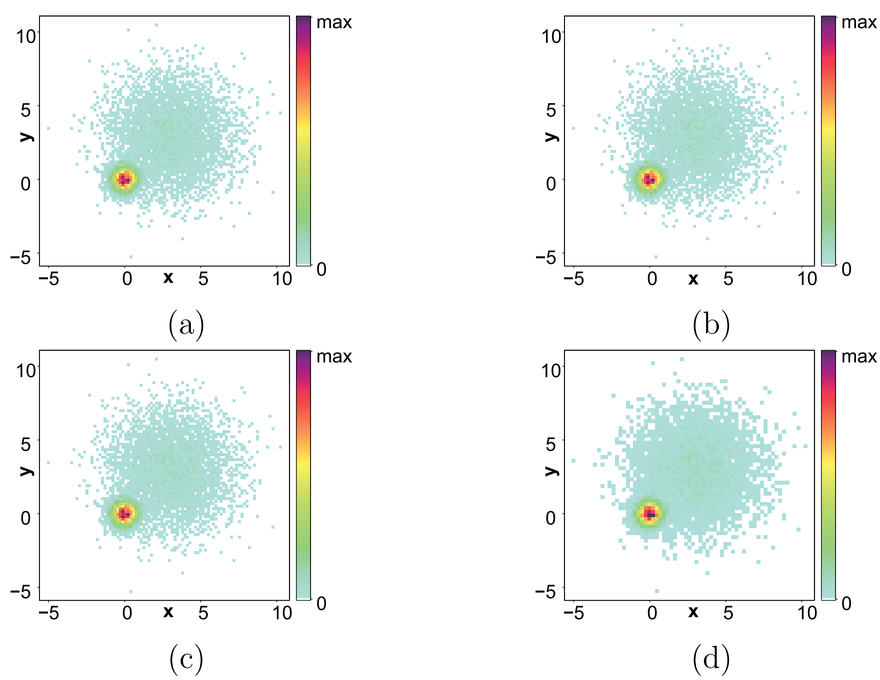

Fig. 4. Image scatter plot of learning weights and parameters of trained models, where the color indicates the density, using the Mian dataset: (a) trained models without an index, (b) trained models with a random index, (c) trained models in order of size, and (d) one of the manually labeled training models.

most representative samples through clustering and according to the differences among domains. Fig. 4 depicts the learning weights and parameters of the trained models when using the proposed method, including the histograms, labels, locations, sigmas, clusters, statistical weights, number of classes, number of visual words, number of clusters, descriptors dimension, directions to the center, location size, and training classes.

The training data must be normalized prior to K-means++ clustering, which is a multi-dimensional normalization technique. The weights can be adjusted on the normalized data to assign weights to each factor. The return value of K-means++ is an array. Values of 0 and 1 represent the distance between the observed value and centroid, respectively, where a smaller distance indicates more closeness to the centroid. Therefore, the accuracy of the results can be evaluated using the Euclidean distance between the factors within the class or the average distance between the measured value and centroid. Experimental results have demonstrated that a larger K value indicates a better clustering effect of K-means++. However, a greater K value is not

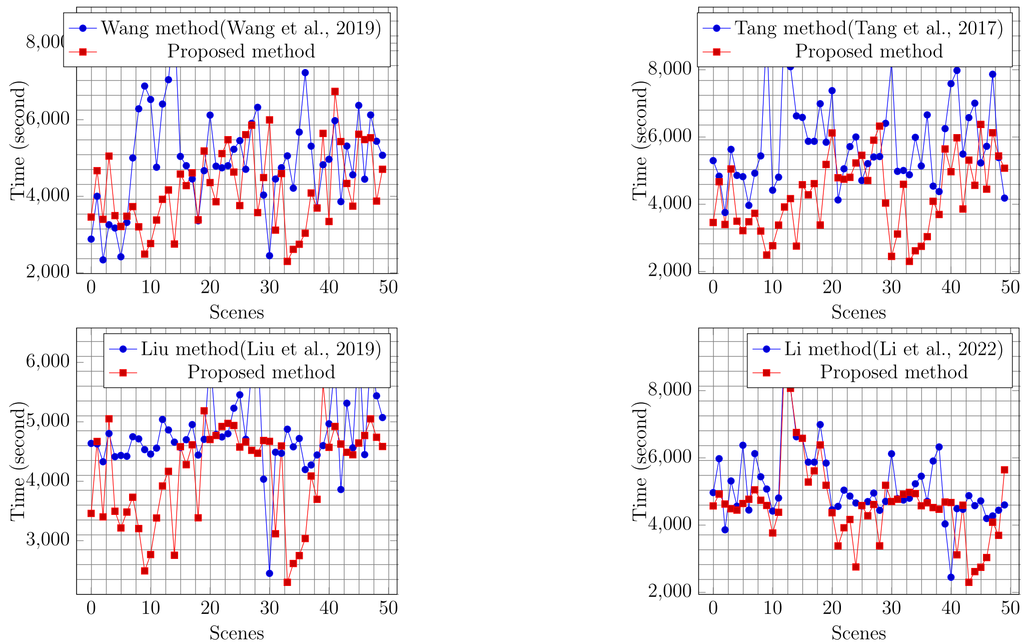

Fig. 5. Results of comparison with proposed method for Mian dataset.

Fig. 6. Experimental results of Mian dataset using proposed method.

always preferable. According to the algorithm, a larger K value results in greater time and space costs for the system, and when the training data are relatively large, excessive K value may cause slow running, thereby wasting substantial time and affecting the efficiency. Finally, in this study, the trained model of the Mian dataset (Mian et al., 2006) was applied to chicken_high, rhino, T-rex_high, parasaurolophus_high, and chef, for the initial ordered values, with manual labeling performed to select the K centroids.

The experimental results are presented in Fig. 6, wherein the dark green objects denote objects to be recognized in the scenes and the red centroids indicate 3D object recognition results. The results of the first experiment indicate that, apart from several misjudgments, no invalid or null judgments appeared. The proposed method exhibited an extremely high recognition rate for the Mian dataset. Fig. 5 presents the results of the 3D object recognition method compared to the methods of Wang et al. (2019), Tang et al. (2017), Liu et al. (2019), and Li et al. (2022). It can be observed that the proposed method achieved better efficiency. Moreover, Table 1 displays the improved 3D IoU of the proposed methods compared to the methods of Wang et al. (2019), Tang et al. (2017), Liu et al. (2019), and Li et al. (2022).

The second experiment performed on the Tosca high-resolution 3D dataset v 1.0 from Bronstein et al. (2008) and Symeonidou et al. (2019), which includes high-resolution 3D implicit shapes in various poses, as illustrated in Fig. 9(a). The dataset contains a total of 80 models, including 11 cats, 6 centaurs, 9 dogs, 4 gorillas, 8 horses, 12 female figures (Victoria), 3 wolves, and two different male figures (David and Michael), incorporating 7 and 20 poses, respectively. As the same class of models has an equal number of vertices, with the same triangulation in a compatible manner, the models could be used as a per-vertex ground truth correspondence in the experiments.

The index and label methods were retained in the second experiment, as the first experiments demonstrated that the trained models

Table 1
Comparative results of 3D IoU based on Mian dataset.

Name	Proposed method	Wang method (Wang et al., 2019)	Tang method (Tang et al., 2017)	Liu method (Liu et al., 2019)	Li method (Li et al., 2022)
cheff	84	80	75.45	63.98	83.98
chicken_high	78.33	76	66.38	72.5	77.5
parasaurolophus_high	82.79	79	78.22	72.56	82.56
rhino_high	82.96	82	75.65	66.58	76.8
T-rex_high	78.44	78	66.52	63.25	73.87

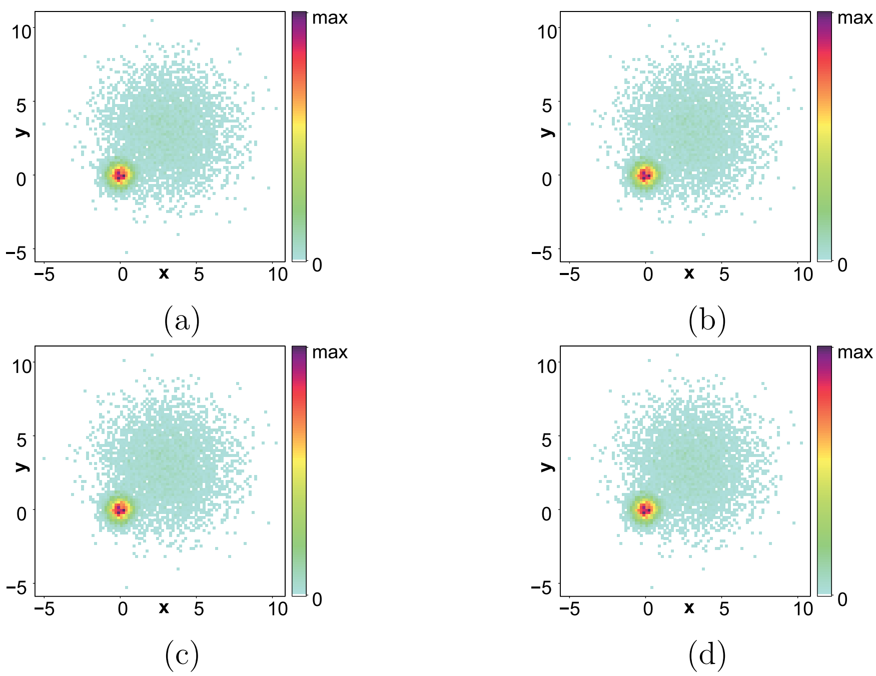

Fig. 7. Image scatter plot of learning weights and parameters of trained models with color indicating density, when using Tosca high-resolution 3D dataset: (a) trained models with random index, (b) trained models in order of size, and (c) and (d) manually labeled training models.

with no index did not provide better effects and yielded a low recognition rate. Fig. 7 depicts the learning weights and parameters of trained models when using the proposed method. Finally, the trained model of the Tosca high-resolution 3D dataset v 1.0 was applied to cat0, centaur3, david12, dog8, gorilla1, horse0, michael15, victoria25, and wolf0 for the initial ordered values with manual indexing and labeling performed to select the K centroids.

The experimental results are presented in Fig. 9(b), in which the dark green objects denote the models to be recognized and the red centroids indicate the 3D object recognition results. It can be observed that the proposed method exhibited an extremely high recognition rate for the Tosca high-resolution 3D dataset. Fig. 8 depicts the results of the 3D object recognition for the proposed method compared to the methods of Wang et al. (2019), Tang et al. (2017), Liu et al. (2019) and Li et al. (2022). This figure also demonstrates that the proposed method achieved higher efficiency. Moreover, Table 2 presents the improved 3D IoU of the proposed methods compared to the methods of Wang et al. (2019), Tang et al. (2017), Liu et al. (2019), and Li et al. (2022).

4. Conclusions and future directions

The aim of this study was to solve the problem of the low recognition rate of 3D implicit shape objects. A semi-supervised active learning hypothesis verification approach for 3D reconstruction in light-field imaging has been proposed. This resolution can contribute to virtual reality fusion in 3D displays. The main advantage of the proposed method is that it does not require additional hardware implementation and it offers robust self-adaptability compared to traditional pattern recognition approaches. Therefore, hardware savings can be achieved, without manual effort and with fewer errors. The proposed method was tested on fixed datasets with many samples and in numerous experiments. The results revealed that the proposed method improved the recognition rate of 3D implicit shape objects and could overcome the shortcomings of local feature clustering. Future experiments can focus on more adaptive classification algorithms, such as the construction and merging of viewpoint feature histograms and the KD-tree structure, to enhance the interactivity of the light-field display. Additionally, in the future, the texture mapping of illumination consistency can be continued on this basis. Thus, combined with the prior knowledge of the scene before the light-field acquisition, through the reverse deduction, reconstruction, and decoding in the subsequent stage, the light-field display stage can obtain images with more optimized resolution and dimensional space information so that the virtual reality fusion effect of the light-field imaging is better.

5. Discussion

In general, the sample is gradually increased in the image processing stage in 3D reconstruction under the temporal and spatial domains. When the semi-supervised active learning method, combined with 3D feature extraction, is not used, the light-field system generally randomly selects samples or uses some manual rules to provide samples to be marked for manual marking. Although this can also bring some improvement, the cost of labeling is always relatively large. The proposed method is to obtain the sample that is "difficult" to classify through active learning, allow the manual reconfirmation and review, and then use the semi-supervised active learning model to train the manually annotated data again, so as to gradually improve the effect of the model and integrate the manual experience into the active learning

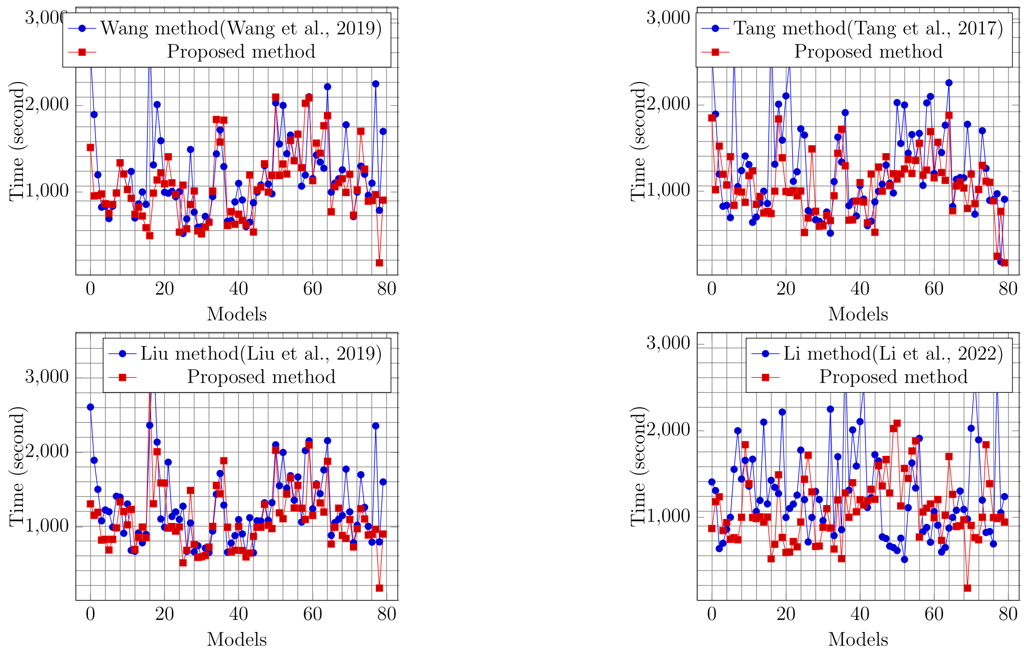

Fig. 8. Results of comparison with proposed method for Tosca high-resolution 3D dataset.

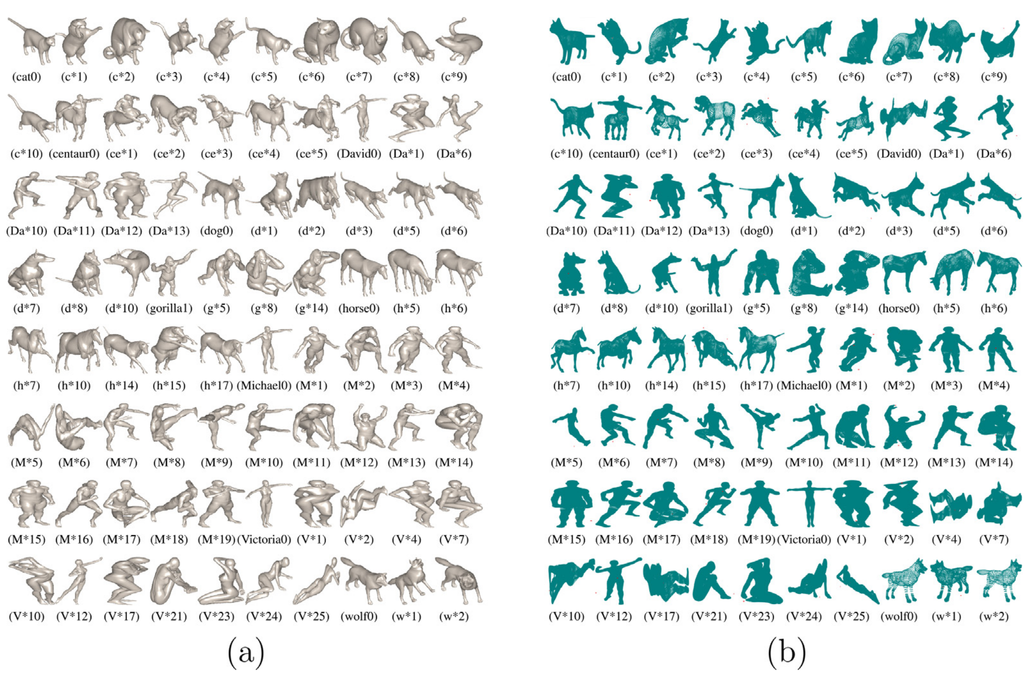

Fig. 9. 80 models of Tosca high-resolution dataset (Bronstein et al., 2008; Symeonidou et al., 2019): (a) source models and (b) experimental results of proposed method.

Table 2
Comparison results 3D IoU based on the Mian dataset.

Name	Proposed method	Wang method (Wang et al., 2019)	Tang method (Tang et al., 2017)	Liu method (Liu et al., 2019)	Li method (Li et al., 2022)
Cat	81.81	63.63	63.63	54.5	80
Centaur	66.67	50	33.33	50	65.55
David	71.43	57.14	57.14	57.14	69
Dog	88.89	66.66	55.56	66.66	78
Gorilla	75	50	56	59	69
Horse	75	62.5	50	62.5	74.59
Michael	75	65	60	63.13	72
Victoria	83.33	66.66	67.66	75	81
Wolf	66.67	43.56	35.33	55.22	65.59

model. The proposed method uses the feature extraction module of the trained model, extracts the features of all labeled data, and according to the classification results, takes the average for the features of each category. Use the classifier of the trained model to classify the average feature and the unlabeled data feature and compare the classification results of the two features. However, there is a lack of priority sorting methods for the selected unlabeled data in the algorithm process. In many cases, particularly in the early stage of training, there may be a large number of mixed features and unlabeled data features with inconsistent classification results, which makes a large number of unlabeled data selected. At this time, a further selection mechanism (perhaps sorting) is required to further filter these data and select the best among the best. The suitability and superiority of the proposed method compared with other state-of-the-art algorithms will be further investigated in future work.

CRediT authorship contribution statement

Zhenhao Wang: 3D modeling methodology, Program development, Writing – original draft, Conceptualization, Supervision, Project administration, Writing – review & editing, Funding acquisition. **Rui Xu:** Writing – review & editing, Funding acquisition. **Tingyuan Nie:** Writing – review & editing. **Dong Xu:** Conceptualization, Writing – original draft, Writing – review & editing, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgments

This work was supported in part by the Natural Science Foundation of Shandong Province, China under Grant ZR2021MF101, in part by the Postdoctoral Science Foundation of China under Grant 2020TQ0350, and in part by the National Natural Science Foundation of China under Grant 62171247.

References

Arthur, D., Vassilvitskii, S., 2006. K-Means++: The Advantages of Careful Seeding. Technical Report, Stanford.
Bletterer, A., Payan, F., Antonini, M., 2020. A local graph-based structure for processing gigantic aggregated 3D point clouds. IEEE Trans. Vis. Comput. Graphics (01), 1.
Bronstein, A.M., Bronstein, M.M., Kimmel, R., 2008. Numerical Geometry of Non-Rigid Shapes. Springer Science & Business Media.
Chen, Z., Zeng, W., Yang, Z., Yu, L., Fu, C.W., Qu, H., 2019. Lassonet: Deep lasso-selection of 3d point clouds. IEEE Trans. Vis. Comput. Graphics 26 (1), 195–204.
Cho, D.-Y., Kang, M.-K., 2021. Human gaze-aware attentive object detection for ambient intelligence. Eng. Appl. Artif. Intell. 106, 104471.
Comaniciu, D., 2003. Nonparametric information fusion for motion estimation. In: 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings, Vol. 2. IEEE Computer Society, p. 59.
Dalal, N., 2006. Finding People in Images and Videos (Ph.D. thesis). Institut National Polytechnique de Grenoble-INPG.
Feng, M., Gilani, S.Z., Wang, Y., Zhang, L., Mian, A., 2020. Relation graph network for 3D object detection in point clouds. IEEE Trans. Image Process. 30, 92–107.
Fu, K., Peng, J., He, Q., Zhang, H., 2021. Single image 3D object reconstruction based on deep learning: A review. Multimedia Tools Appl. 80 (1), 463–498.
Gao, C., Peng, Y., Wang, R., Zhang, Z., Li, H., Liu, X., 2021. Foveated light-field display and real-time rendering for virtual reality. Appl. Opt. 60 (28), 8634–8643.
Glover, J., Popovic, S., 2013. Bingham procrustean alignment for object detection in clutter. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, pp. 2158–2165.
Kim, M., Lee, K., Han, Y., Lee, J., Nam, B., 2021. Generating 3D texture models of vessel pipes using 2D texture transferred by object recognition. J. Comput. Des. Eng. 8 (1), 475–487.
Li, L., Fu, H., Ovsjanikov, M., 2022. WSDesc: Weakly supervised 3D local descriptor learning for point cloud registration. IEEE Trans. Vis. Comput. Graphics.
Liu, H., Cong, Y., Yang, C., Tang, Y., 2019. Efficient 3D object recognition via geometric information preservation. Pattern Recognit. 92, 135–145.
Liu, D., Zhang, Y., Luo, L., Li, J., Gao, X., 2021. PDC-net: robust point cloud registration using deep cyclic neural network combined with PCA. Appl. Opt. 60 (11), 2990–2997.
MacQueen, J., et al., 1967. Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Oakland, CA, USA, Vol. 1. pp. 281–297.
Mian, A.S., Bennamoun, M., Owens, R., 2006. Three-dimensional model-based object recognition and segmentation in cluttered scenes. IEEE Trans. Pattern Anal. Mach. Intell. 28 (10), 1584–1601.
Pan, Y., 2021. On visual understanding. Front. Inf. Technol. Electron. Eng. 1–3.
Peng, B., Wang, W., Dong, J., Tan, T., 2021. Learning pose-invariant 3D object reconstruction from single-view images. Neurocomputing 423, 407–418.
Prakhya, S.M., Liu, B., Lin, W., 2016. Detecting keypoint sets on 3D point clouds via histogram of normal orientations. Pattern Recognit. Lett. 83, 42–48.
Ren, H., Ni, L., Li, H., Sang, X., Gao, X., Wang, Q., 2020. Review on tabletop true 3D display. J. Soc. Inf. Disp. 28 (1), 75–91.
Rusu, R.B., Blodow, N., Beetz, M., 2009. Fast point feature histograms (FPFH) for 3D registration. In: 2009 IEEE International Conference on Robotics and Automation. IEEE, pp. 3212–3217.
Rusu, R.B., Blodow, N., Marton, Z.C., Beetz, M., 2008. Aligning point cloud views using persistent feature histograms. In: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, pp. 3384–3391.
Song, W., Li, D., Sun, S., Zhang, L., Xin, Y., Sung, Y., Choi, R., 2022. 2D&3DHNet for 3D object classification in LiDAR point cloud. Remote Sens. 14 (13), 3146.
Symeonidou, A., Kizhakkumkara, R.M., Birnbaum, T., Schelkens, P., 2019. Efficient holographic video generation based on rotational transformation of wavefields. Opt. Express 27 (26), 37383–37399.
Tang, Z., Chen, G., Han, Y., Liao, X., Ru, Q., Wu, Y., 2022. Bi-stage multi-modal 3D instance segmentation method for production workshop scene. Eng. Appl. Artif. Intell. 112, 104858.
Tang, K., Song, P., Chen, X., 2017. 3D object recognition in cluttered scenes with robust shape description and correspondence selection. IEEE Access 5, 1833–1845.
Wang, C., Shu, Q., Yang, Y., Yuan, F., 2018. Point cloud registration in multidirectional affine transformation. IEEE Photonics J. 10 (6), 1–15.
Wang, J., Yang, Y., Shao, M., Zhou, Y., 2020b. Three-dimensional measurement for rigid moving objects based on multi-fringe projection. IEEE Photonics J. 12 (4), 1–14.
Wang, C., Yang, Y., Shu, Q., Yu, C., Cui, Z., 2020a. Point cloud registration algorithm based on Cauchy mixture model. IEEE Photonics J. 13 (1), 1–13.
Wang, C., Zhang, Q., Ma, B., Xia, Z., Li, J., Luo, T., Li, Q., 2022. Light-field image watermarking based on geranion polar harmonic Fourier moments. Eng. Appl. Artif. Intell. 113, 104970.
Wang, Z., Zhao, Y., Wang, S., 2019. Approach for improving efficiency of three-dimensional object recognition in light-field display. Opt. Eng. 58 (12), 123101.
Yang, Y., Zhuang, Y., Pan, Y., 2021. Multiple knowledge representation for big data artificial intelligence: framework, applications, and case studies. Front. Inf. Technol. Electron. Eng. 22 (12), 1551–1558.
Zhang, Z., Hu, L., Deng, X., Xia, S., 2020. Weakly supervised adversarial learning for 3D human pose estimation from point clouds. IEEE Trans. Vis. Comput. Graphics 26 (5), 1851–1859.
Zhu, Z., You, D., Zhou, F., Wang, S., Xie, Y., 2021. Rapid 3D reconstruction method based on the polarization-enhanced fringe pattern of an HDR object. Opt. Express 29 (2), 2162–2171.