Multimedia Tools and Applications (2021) 80:26121–26143
https://doi.org/10.1007/s11042-021-10794-3

Novel methods for noisy 3D point cloud based object recognition

Xian-Feng Han[1] · **Xin-Yu Yan**[2] · **Shi-Jie Sun**[3]

Received: 28 May 2020 / Revised: 22 December 2020 / Accepted: 10 March 2021 /
Published online: 28 April 2021

Abstract

3D point cloud based object recognition becomes increasingly important in the last few years, as the widely use of point cloud over the low-cost 3D sensors have developed rapidly. However, the obtained 3D point cloud is inevitably contaminated with noise due to physical and environmental factors, which has a negative impact on recognition task. To address this problem, a complete object recognition framework for 3D noisy point cloud is presented into which a pre-processing step of filtering is integrated for the first time. In the filtering phase, our two proposed approaches, named Guided 3D Point Cloud Filter (G3DF) and Iterative Guidance Normal Filter (IGNF), are taken into account to produce high-quality point cloud model. Then, on the basis of advantages of local-based and global-based descriptors, a new type of feature descriptor, called Local-to-Global Histogram (LGH), is proposed, which contains Local Viewpoint Feature Histogram (LVFH) and Local Ensemble of Shape Function (LESF). Experimental results show that the comprehensive classification performance yielded by using proposed filters and descriptors is competitive compared to other state-of-the-art combinations. In particularly, the composition of G3DF and LVFH is more suited for real-time applications.

Keywords 3D point cloud · Object recognition · Noise reduction · G3DF · IGNF · LVFH · LESF

✉ Xian-Feng Han
xianfenghan@swu.edu.cn

Xin-Yu Yan
xinyuyan@tju.edu.cn

Shi-Jie Sun
shijiesun@chd.edu.cn

1 College of Computer and Information Science, Southwest University, Chongqing, 400715, China

2 Tianjin University, Tianjin, 300072, China

3 Chang'an University, Xi'an, ShaanXi, 710064, China

1 Introduction

Object recognition [28] based on 3D point cloud is arguably a fundamental component of numerous real-world applications, such as multimedia [30, 33], autonomous driving [7], augmented/virtual reality [37] and robotics [11]. The aim of this field is to identify the object in the point cloud model correctly, which has been drawn increasing attention recently. The booming development of this research field is mainly put down to the following factors: 1) The advent of new-generation 3D data acquisition sensors (e.g. Microsoft Kinect [6], Time of Flight [14]) makes 3D point cloud become increasingly popular [21]. 2) The development of high-performance computing devices accelerates the running speed of computationally intensive 3D object recognition system. 3) Compared to 2D images, 3D point cloud provides much richer geometric information which contributes to improving the representation ability for characterizing objects [9]. 4) Furthermore, features are extracted from 3D data can address many issues encountered by that from 2D image due to the characteristic [8] of 3D point cloud.

In [1], the authors showed pipelines for local-based and global-based 3D object recognition in scenes, where local-based pipeline is mainly composed of keypoint extraction, description, matching, correspondence grouping and hypothesis verification. While the global-based pipeline includes four steps: segmentation, description, matching and hypothesis verification. Guo et al. [35] gave a review of methods used in each stage of local-based object recognition pipeline. Alexandre [5] used another type of pipeline for both local and global-based object recognition, which consists of keypoint detection, descriptor extraction and matching process. Different from these previous works, 1) we categorize both local-based and global-based object recognition pipeline into two stages: training and testing phase. 2) We introduce the machine learning algorithm to accomplish learning and prediction tasks. 3) It is worth noting that we integrate pre-processing (filtering) into our framework. 4) In terms of application, we use our framework to implement single object recognition task instead of scene-based object recognition.

For filtering, Han et al. [14] classified the existing filtering methods into seven groups, namely, statistical-based, neighborhood-based, projection-based PDEs-based, signal processing-based, hybrid methods and others. However, it can be seen from the discussion that most of these works are time-consuming. About 3D descriptor [34, 36] gave a comprehensively insightful investigation of the existing 3D point cloud descriptors, which are divided into two categories: hand-crafted based and deep-learning based methods. Although the deep-learning approaches currently achieve far better performance compared to hand-crafted ways, lack of large-scale training dataset and the sparsity of 3D point cloud challenged deep learning strategies .

In this paper, a complete framework is devised for noisy 3D point cloud based object recognition, which takes the pre-processing into account, as shown in Fig. 1. And two of our proposed filters, called Guided 3D Point Cloud Filter and Iterative Guidance Normal Filter, will be considered as the major pre-processing algorithms. Furthermore, motivated by the advantages of local-based and global-based descriptors, a new kind of point cloud based descriptor, named Local-to-Global Histogram is proposed. Experimental results demonstrate that the designed descriptors achieve high-level performance in terms of descriptiveness, robustness and efficiency. And the integration of filter algorithms into our framework actually contribute to improving the recognition performance.

The main contributions of our work are as follows:

Fig. 1 Block diagrams of general framework for noisy 3D point cloud based object recognition

1. To the best of our knowledge, this is the first paper that takes the pre-processing step into account to formulate a complete noisy 3D point cloud based object recognition framework.
2. This paper proposes the Local-to-Global Histogram descriptor (LGH), including Local Viewpoint Feature Histogram (LVFH) and Local Ensemble of Shape Function (LESF).
3. Comprehensive experiments are carried out to verify the roles of our filters and descriptors in the noisy 3D point cloud based object recognition framework.

The remainder of this paper is organized as follows. Section 2 simply introduce the background of our work. Section 3 briefly outlines our proposed filters G3DF and IGNF. Section 4 describes the Local-to-Global Histogram descriptors. Comparative and comprehensive experiments are conducted in Section 5. Section 6 gives the demonstration of proposed algorithms in our 3D object recognition pipeline. Section 7 draws the conclusion of this paper.

2 Background

Without loss of generality, the noisy 3D point cloud based recognition processing pipeline is distinctly divided into two broad categories: local-based and global-based framework on the grounds of the type of features used (local based or global based descriptor[10]). Figures 2 and 3 show local-based and global-based pipeline, respectively.

At the conceptual level, the critical blocks of local-based pipeline include 3D keypoint detection, local-based descriptor and matching. The input 3D point cloud model is fed into the 3D keypoints detection [31] phase first to identify points which are distinctive and repeatable enough to highly characterize surface. Then, *local-based descriptor* [22] is constructed by resorting to the geometrical information of the local neighborhood associated with each keypoint. During the matching stage, a kind of classifier is used to classify

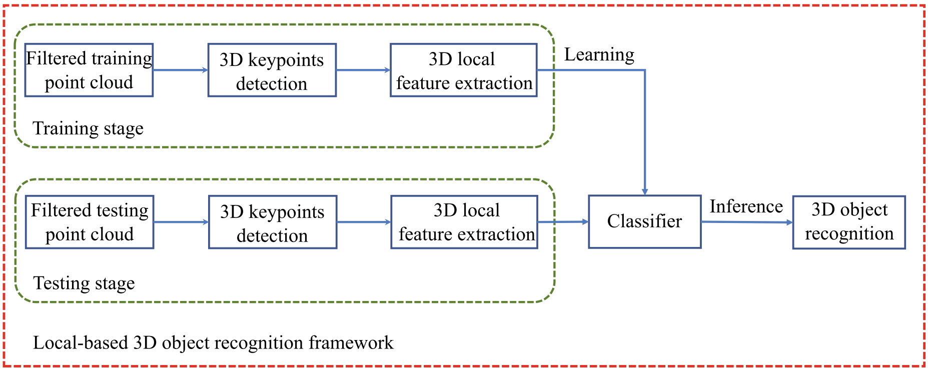


Fig. 2 Block diagrams of local-based methods for noisy 3D point cloud based object recognition

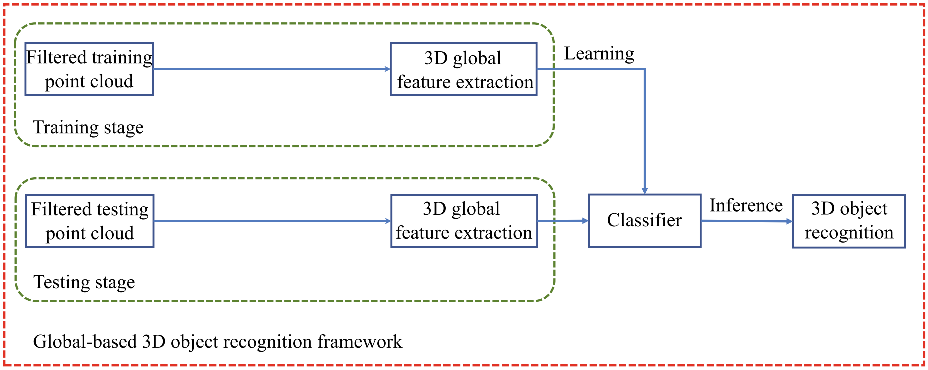


Fig. 3 Block diagrams of global-based framework for noisy 3D point cloud based object recognition

features depending on the patterns learned from training step. In the case of global-based pipeline, since the *global-based descriptor* [19] generally estimates a single descriptor vector encoding the entire geometry of input point cloud, this pipeline therefore only contains global-based descriptor extraction and matching phases.

3 Pre-processing

The raw 3D point cloud models are inevitably contaminated by noise due to physical factors, limitations of sensors. Therefore, in order to obtain high-quality and high-accuracy point cloud model, it is necessary to incorporate the pre-processing operations into the 3D point cloud based object recognition framework. In this section, two of our proposed filters, namely, Guided 3D Point Cloud Filter and Iterative Guidance Normal Filter, will be introduced here.

3.1 Guided 3D point cloud filtering

To reduce the computational cost when filtering point cloud model with huge numbers of points, inspired by Guided Image Filtering [15], the Guided 3D Point Cloud Filtering (G3DF) is proposed in [12], which will be sketched out in this section.

Given a raw input point cloud $P = \{\boldsymbol{p}_i \in R^3\}$, a kdtree structure is constructed to help search the corresponding neighborhood $N(\boldsymbol{p}_i) = \{\boldsymbol{p}_{ij} \in P\}$ for each point $\boldsymbol{p}_i$. And an assumption that there exists a linear relationship between filtered output and guidance point cloud (Here, the input point cloud is considered as guidance) is made in each neighborhood. To be specific, the resulting point is achieved by performing a linear transformation with respect to the corresponding point $\boldsymbol{p}_{ij}$ in $N(\boldsymbol{p}_i)$.

$$\boldsymbol{p}'_{ij} = a_i \boldsymbol{p}_{ij} + \boldsymbol{b}_i \tag{1}$$

Where $\boldsymbol{p}'_{ij}$ refers to the filtered point. a_i and $\boldsymbol{b}_i$ are the coefficients of linear model.

Then, a new cost function measuring the difference between the guidance point and filtered point is defined as follows:

$$J(a_i, \boldsymbol{b}_i) = \sum_{\boldsymbol{p}_{ij} \in N(\boldsymbol{p}_i)} \left((a_i \boldsymbol{p}_{ij} + \boldsymbol{b}_i - \boldsymbol{p}_{ij})^2 + \varepsilon a_i^2\right) \tag{2}$$

Where ε is a regularization parameter to control the filtering effect by preventing a_i from being too large. These coefficients can be got by minimizing (2).

$$a_i = \frac{\left(\frac{1}{|N(\boldsymbol{p}_i)|}\sum \boldsymbol{p}_{ij} \cdot \boldsymbol{p}_{ij} - \bar{\boldsymbol{p}}_i \cdot \bar{\boldsymbol{p}}_i\right)}{\left(\frac{1}{|N(\boldsymbol{p}_i)|}\sum \boldsymbol{p}_{ij} \cdot \boldsymbol{p}_{ij} - \bar{\boldsymbol{p}}_i \cdot \bar{\boldsymbol{p}}_i\right) + \varepsilon} \tag{3}$$

$$\boldsymbol{b}_i = \bar{\boldsymbol{p}}_i - a_i \cdot \bar{\boldsymbol{p}}_i \tag{4}$$

Where

$$\bar{\boldsymbol{p}}_i = \frac{1}{|N(\boldsymbol{p}_i)|} \sum_{\boldsymbol{p}_{ij} \in N(\boldsymbol{p}_i)} \boldsymbol{p}_{ij} \tag{5}$$

Once estimated a_i and $\boldsymbol{b}_i$, the filtered point $\boldsymbol{p}_i'$ corresponding to the present query point $\boldsymbol{p}_i$ in its own neighborhood $N(\boldsymbol{p}_i)$ can be computed.

$$\boldsymbol{p}_i' = a_i \boldsymbol{p}_i + \boldsymbol{b}_i \tag{6}$$

Finally, the resulting point cloud model is obtained by performing the linear transformation for every point. This algorithm is discussed in more detail in [12]

Algorithm 1: Guided 3D Point Cloud Filtering.

Input: raw point cloud P with N points, ε
Output: Filtered point cloud $\hat{P}$

```
Build kdtree structure for point cloud;
for i = 1 to N do
    Calculate the neighborhood p_ij of p_i;
    Computer the number of neighbors K = ||p_ij||;
    p̄_i = (1/K) Σ p_ij   a_i = ((1/K) Σ p_ij·p_ij − p̄_i·p̄_i) / ((1/K) Σ p_ij·p_ij − p̄_i·p̄_i) + ε);
    b_i = p̄_i − a_i · p̄_i;
    p'_i = a_i p_i + b_i;
end
P̂ = {p'_i};
```

3.2 Iterative guidance normal filter

Since the traditional Bilateral filter and its variations usually cannot achieve satisfactory results by performing simply one single iteration, especially when working with point clouds with different noise levels. Therefore, motivated by [32] and [23], the Iterative Guidance Normal Filter (IGNF for short) is proposed, which consists of three crucial steps: normal estimation, iterative guidance normal filtering and point updating.

3.2.1 Normal estimation

Similar to G3DF algorithm, a kdtree structure is used to find out the local neighborhood $N(\boldsymbol{p}_i) = \{\boldsymbol{p}_{ij} \in P\}$ for each point $\boldsymbol{p}_i$. Depending on this neighborhood, a covariance matrix of size 3×3 is calculated for $\boldsymbol{p}_i$:

$$C_i = \sum_{\boldsymbol{p}_{ij} \in N_{\boldsymbol{p}_i}} e^{\frac{\|\boldsymbol{p}_{ij} - \bar{\boldsymbol{p}}_i\|^2}{\sigma^2}} (\boldsymbol{p}_{ij} - \bar{\boldsymbol{p}}_i)(\boldsymbol{p}_{ij} - \bar{\boldsymbol{p}}_i)^T \tag{7}$$

Where $\bar{\boldsymbol{p}}_i$ is computed in the same way as (5). Here, Gaussian weight instead of uniform weight is adopted to guarantee sharp features.

Due to the fact that C_i is a symmetric positive semi-definite matrix, the eigenvalues $\{\lambda_1, \lambda_2, \lambda_3\}$ corresponding to the eigenvectors $\{\nu_1, \nu_2, \nu_3\}$ are estimated by using (8). Here, assuming that $\lambda_1 < \lambda_2 < \lambda_3$, the eigenvector ν_1 corresponding to the smallest eigenvalue is taken as the normal $\boldsymbol{n}_i$ of $\boldsymbol{p}_i$.

$$C_i \cdot \nu_k = \lambda_k \cdot \nu_k \tag{8}$$

3.2.2 Iterative guidance normal filter

Here, according to the normals and spatial positions of query point $\boldsymbol{p}_i$ and its neighbors $\boldsymbol{p}_{ij}$, a normal-based Bilateral filter is defined in the following form,

$$\boldsymbol{n}_i' = \frac{1}{K_i} \sum_{\boldsymbol{p}_{ij} \in N_{\boldsymbol{p}_i}} exp(-\frac{\|\boldsymbol{p}_i - \boldsymbol{p}_{ij}\|}{2\sigma_s^2} - \frac{\|\boldsymbol{n}_i - \boldsymbol{n}_{ij}\|}{2\sigma_r^2}) \cdot \boldsymbol{n}_i \tag{9}$$

Where,

$$K_i = \sum_{\boldsymbol{p}_{ij} \in N_{\boldsymbol{p}_i}} exp(-\frac{\|\boldsymbol{p}_i - \boldsymbol{p}_{ij}\|}{2\sigma_s^2} - \frac{\|\boldsymbol{n}_i - \boldsymbol{n}_{ij}\|}{2\sigma_r^2}) \tag{10}$$

To handle different level noise well, the iterative idea is introduced to (9) to form the iterative guidance normal filter as follows:

$$\boldsymbol{n}_i^{k+1} = \frac{1}{K_i^k} \sum_{\boldsymbol{p}_{ij} \in N_{\boldsymbol{p}_i}} exp(-\frac{\|\boldsymbol{p}_i - \boldsymbol{p}_{ij}\|}{2\sigma_s^2} - \frac{\left\|\boldsymbol{n}_i^k - \boldsymbol{n}_{ij}^k\right\|}{2\sigma_r^2}) \cdot \boldsymbol{n}_i^k \tag{11}$$

Where,

$$K_i^k = \sum_{\boldsymbol{p}_{ij} \in N_{\boldsymbol{p}_i}} exp(-\frac{\|\boldsymbol{p}_i - \boldsymbol{p}_{ij}\|}{2\sigma_s^2} - \frac{\left\|\boldsymbol{n}_i^k - \boldsymbol{n}_{ij}^k\right\|}{2\sigma_r^2}) \tag{12}$$

3.2.3 Point updating

After performing normal filtering operation, the point positions should be updated to match the filtered normals accordingly. The updating scheme proposed by Sun et al. [29] is modified to obtain the new point positions by iteratively operating the following equation,

$$\hat{\boldsymbol{p}}_i^{k+1} = \hat{\boldsymbol{p}}_i^k + \frac{1}{\left|N_{\boldsymbol{p}_i}\right|} \sum \hat{\boldsymbol{n}}_i \left[\hat{\boldsymbol{n}}_{ij} \left(\hat{\boldsymbol{p}}_{ij}^k - \hat{\boldsymbol{p}}_i^k\right)\right] \tag{13}$$

See paper [13] for more information about this algorithm.

Algorithm 2: Iterative Guidance Normal Filtering.

Input: raw point cloud P with N points, σ_r, n_{iter}, p_{iter}
Output: Filtered point cloud $\hat{P}$

Create a kdtree for p;
for *i=1 to N* **do**
  Search the neighborhood $N_{\boldsymbol{p}_i}$ for each point $\boldsymbol{p}_i$;
  Computer surface normal $\boldsymbol{n}_i$ at each point $\boldsymbol{p}_i$;
  Initialize $\boldsymbol{n}_o = \boldsymbol{n}_i$;
  for *k=0 to* $n_{iter} - 1$ **do**
    for *j=1 to* $\left|N_{\boldsymbol{p}_i}\right|$ **do**
      $s = \left\|\boldsymbol{p}_{ij} - \boldsymbol{p}_i\right\|, r = \left\|\boldsymbol{n}_{ij}^k - \boldsymbol{n}_i^k\right\|$;
      $w_s = exp(-\frac{s^2}{\sigma_s}), w_r = exp(-\frac{r^2}{\sigma_r})$;
      $\boldsymbol{n}_{sum} += w_s w_r \boldsymbol{n}_j^k, normalize += w_s w_r$;
    end
    $\boldsymbol{n}_i^{k+1} = \boldsymbol{n}_{sum}/normalize$;
  end
  for *m=0 to* $p_{iter} - 1$ **do**
    for *j=1 to* $\left|N_{\boldsymbol{p}_i}\right|$ **do**
      $\boldsymbol{p}_{sum} += \boldsymbol{n}_i^{n_{iter}}[\boldsymbol{n}_{ij}^{n_{iter}}(\boldsymbol{p}_{ij}^m - \boldsymbol{p}_i^m)]$;
    end
    $\boldsymbol{p}_i^{m+1} += \boldsymbol{p}_i^m + \boldsymbol{p}_{sum}/\left|N_{\boldsymbol{p}_i}\right|$;
  end
end
$\hat{P} = \left\{p_i^{p_{iter}}\right\}$;

4 Local-to-global histogram

How to design suitable feature descriptors for 3D point cloud based object recognition is another considerably important issue. In this section, a new kind of descriptor, named Local-to-Global Histogram (LGH), is put forward, which is on the basis of analysis of advantages of local-based and global-based features. The basic idea behind LGH is that a global-based descriptor is built over the sub point cloud determined by the keypoint and its neighbors. In the following, Local Viewpoint Feature Histogram and Local Ensemble of Shape Function will be depicted respectively.

4.1 Local viewpoint feature histogram

For 3D point cloud, surface normal and curvature are two fundamental attributes of point. However, they are difficult to capture much richer geometric information [25]. On the other hand, the relationships among neighboring points play an important role in getting better surface variation. Therefore, the Local Viewpoint Feature Histogram is based on sub point cloud and relationships determined by one keypoint and its corresponding neighbors. Given an input point cloud $P = \left\{\boldsymbol{p}_i \in R^3\right\}$, the process of building LVFH descriptor consists of the following steps:

In the first stage, a 3D keypoint detection algorithm (e.g. SIFT, Harris3D) is used to select keypoints from the input point cloud. For each keypoint $\boldsymbol{p}_i$, the r-ball strategy is adopted to search $\boldsymbol{p}_i's$ corresponding neighbors $\boldsymbol{p}_j \in \boldsymbol{P}$ which lies within the sphere support region defined by r. These neighboring points together with keypoint can be thought of as a sub point cloud P_i from the entire model.

The second stage is the construction of global-based feature on the sub point cloud mentioned above. First, for each neighboring point $\boldsymbol{p}_j$, a local reference frame (LRF) shown in Fig. 4 is built grounded on the present keypoint $\boldsymbol{p}_i$ and its corresponding surface normal $\boldsymbol{n}_i$. The specific definition of the LRF is as follows:

$$\boldsymbol{x} = \boldsymbol{n}_{p_i}, \tag{14}$$

$$\boldsymbol{y} = \left(\boldsymbol{p}_j - \boldsymbol{p}_i\right) \times \boldsymbol{x}, \tag{15}$$

$$\boldsymbol{z} = \boldsymbol{x} \times \boldsymbol{y} \tag{16}$$

The reason why this kind of reference frame is employed is that 1) inspiration of the successful employment of LRF for Point Feature Histograms descriptor to depict the relationship between point pairs, 2) the introduction of this LRF can transform the relationship between points into quantitatively angles to reduce the number of features. (See Fig. 5)

Once the reference frame is accomplished, three angular features α, β, θ and a distance d between keypoint $\boldsymbol{p}_i$ and its neighbor $\boldsymbol{p}_j$ are subsequently estimated using the following equations.

$$\alpha = \boldsymbol{y} \cdot \boldsymbol{n}_j \tag{17}$$

$$\beta = \left(\boldsymbol{x} \cdot \left(\boldsymbol{p}_j - \boldsymbol{p}_i\right)\right) / \left\|\boldsymbol{p}_j - \boldsymbol{p}_i\right\| \tag{18}$$

$$\theta = arctan(\boldsymbol{z} \cdot \boldsymbol{n}_j, \boldsymbol{x} \cdot \boldsymbol{n}_j) \tag{19}$$

$$d = \|\boldsymbol{p}_i - \boldsymbol{p}_j\| \tag{20}$$

Where the first three features α, β and θ are utilized to measure the difference between normals for points. The fourth feature d is a measurement of the distance between points. These four features are translation and rotation invariant [24, 26].

This procedure will be performed for every pair of point $\boldsymbol{n}_i$'s neighbors and $\boldsymbol{n}_i$ in this sub point cloud. These four features are then constructed into four corresponding 45-bin histograms, respectively.

The fifth angular feature of Local Viewpoint Feature Histogram is a viewpoint-dependent component similar to the viewpoint feature histogram (VFH). In our approach, we calculate the angle φ between the keypoint $\boldsymbol{p}_i$ (as viewpoint direction vector) and its neighbor $\boldsymbol{p}_j$'s

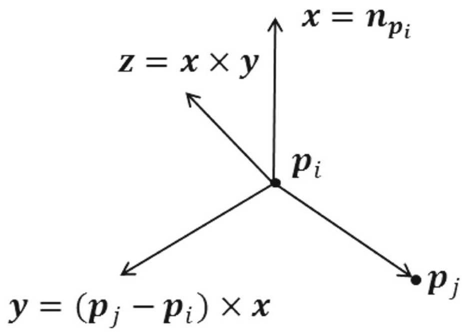


Fig. 4 The local reference frame

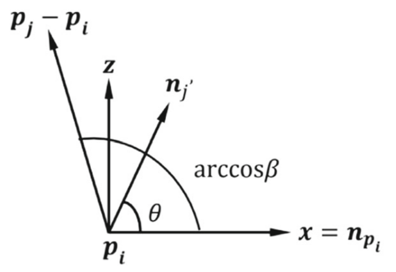


Fig. 5 The features for local viewpoint feature histograms

normal $\boldsymbol{n}_j$. And this feature is subsequently divided into a 128-bin histogram.

$$\varphi = arccos\left(\boldsymbol{n}_j \cdot \boldsymbol{p}_i / \left\|\boldsymbol{p}_i\right\|\right) \tag{21}$$

The final form of the local viewpoint feature histogram is the concatenation of these five histograms, i.e. a five tuple $\{\alpha, \beta, \theta, d, \varphi\}$. Figure 6 gives an example of the LVFH feature extracted from one support region on an apple's point cloud model.

4.2 Local ensemble of shape function

The second histogram descriptor based on the Local-to-Global strategy is the Local Ensemble of Shape Function (LESF). The principal idea behind the LESF is that we extract the global-based feature descriptor Ensemble of Shape Function from sub point cloud defined by the spherical neighborhood of each keypoint. The establishment of LESF descriptor consists of the following stages.

The first phase is similar to the construction of LVFH, where 3D keypoint detector is chosen to define keypoints from point cloud model and determine the corresponding sub point clouds.

The second phase is to build global ensemble of shape function features on each sub point cloud based on the shape function distributions, which contains ten 64-bin histograms. The major steps are summarized as follows:

First, for each sub point cloud P_i, a computationally efficient method is used to create a voxel grid of size $64 \times 64 \times 64$ which can be treated as a rough approximation of the

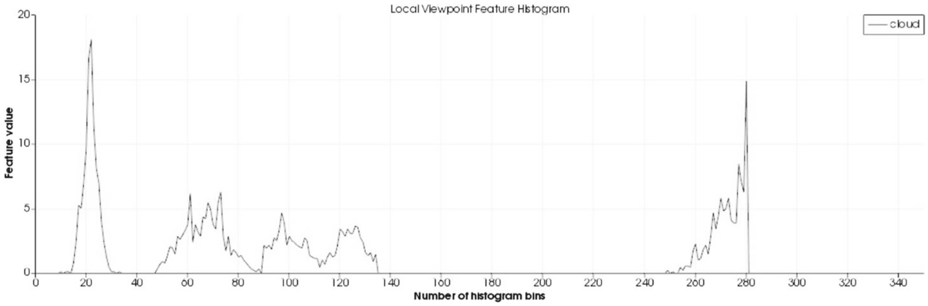


Fig. 6 Example of LVFH defined on the spherical neighborhood around one keypoint on an apple's point cloud model

surface of the sub point cloud. This operation aims to assist in helping complete the task of classification of attributes involved in the subsequent processing.

Second, three points $\boldsymbol{p}_{i1}, \boldsymbol{p}_{i2}, \boldsymbol{p}_{i3}$ are randomly selected from the sub point cloud P_i. And three important shape functions are estimated using corresponding equations. The first one is D2 function which is capable of describing the entire geometry of the sub point cloud and distinguishing rough shapes. However, it does not work well with incomplete point cloud because of the fact that D2 function has limited expressive ability.

$$d_{i12} = \left\| \boldsymbol{p}_{i1} - \boldsymbol{p}_{i2} \right\| \tag{22}$$

$$d_{i13} = \left\| \boldsymbol{p}_{i1} - \boldsymbol{p}_{i3} \right\| \tag{23}$$

$$d_{i23} = \left\| \boldsymbol{p}_{i2} - \boldsymbol{p}_{i3} \right\| \tag{24}$$

The second one is A3 function that is defined as the angle between two lines determined by three points. A3 function encapsulates the features of sub point cloud from a different aspect to enhance the expression ability of descriptor, which is scale invariant. The definition of A3 function is yielded as follows:

$$cos\theta_1 = \left| \frac{(\boldsymbol{p}_{i1} - \boldsymbol{p}_{i2}) \cdot (\boldsymbol{p}_{i1} - \boldsymbol{p}_{i3})}{\left\| \boldsymbol{p}_{i1} - \boldsymbol{p}_{i2} \right\| \left\| \boldsymbol{p}_{i1} - \boldsymbol{p}_{i3} \right\|} \right| \tag{25}$$

$$cos\theta_2 = \left| \frac{(\boldsymbol{p}_{i1} - \boldsymbol{p}_{i2}) \cdot (\boldsymbol{p}_{i2} - \boldsymbol{p}_{i3})}{\left\| \boldsymbol{p}_{i1} - \boldsymbol{p}_{i2} \right\| \left\| \boldsymbol{p}_{i2} - \boldsymbol{p}_{i3} \right\|} \right| \tag{26}$$

$$cos\theta_3 = \left| \frac{(\boldsymbol{p}_{i1} - \boldsymbol{p}_{i3}) \cdot (\boldsymbol{p}_{i2} - \boldsymbol{p}_{i3})}{\left\| \boldsymbol{p}_{i1} - \boldsymbol{p}_{i3} \right\| \left\| \boldsymbol{p}_{i2} - \boldsymbol{p}_{i3} \right\|} \right| \tag{27}$$

The third one is D3 function, which is defined as area of the region enclosed by three points as follow:

$$p = d_{i12} + d_{i13} + d_{i23} \tag{28}$$

$$S = \frac{1}{2} p * (p - d_{i12}) * (p - d_{i13}) * (p - d_{i23}) \tag{29}$$

These three functions are selected due to the fact that they have simple form and are easy to compute. More importantly, they are invariant to small perturbation caused by noise.

Third, 3D Bresenham line generation algorithm is put into use to trace three lines determined by points $\boldsymbol{p}_{i1}, \boldsymbol{p}_{i2}, \boldsymbol{p}_{i3}$. These lines are then classified into three categories: On the surface of sub point cloud, OFF the surface of sub point cloud, part on and part off the surface of sub point cloud (MIXED).

Fourth, according to the locations of lines, the values of D2 function, A3 function and D3 function are divided into ON, OFF and MIXED cases. The corresponding values are placed into associated histograms respectively, totally nine histograms. Particularly, the ratio of values of D2 function belonging to the MIXED type is employed to construct one more shape histogram.

Fifth, the concatenation of these ten histograms formulates the final LESF descriptor for the sub point cloud P_i

Figure 7 is an example of LESF histogram feature extracted from one sub point cloud defined on a banana's point cloud model.

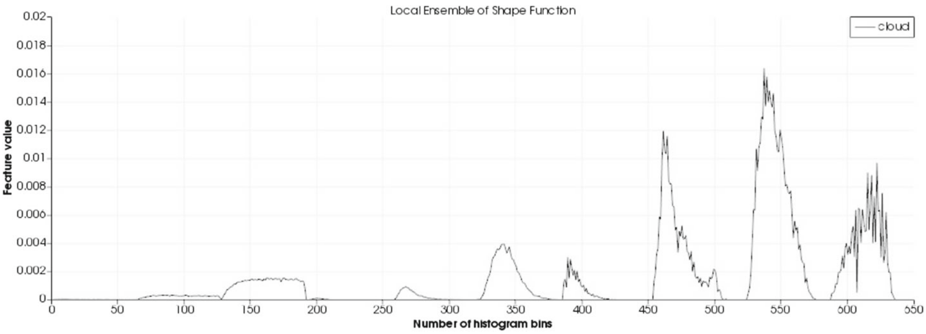


Fig. 7 Example of LESF extracted from sub point cloud defined around one keypoint on a banana's point cloud model

5 Experimental setup and performance evaluation

After introduction of proposed filters and descriptors, comprehensive experiments and comparisons are made to illustrate their performance in terms of descriptiveness, robustness and efficiency.

5.1 Dataset

The experimental results on the publicly available dataset, named Washington RGB-D Objects Dataset [20], are shown. This dataset comes from http://rgbd-dataset.cs.washington.edu/dataset/rgbd-dataset_pcd_ascii/, involving 207,621 3D point clouds (in PCD format) of view of 300 objects which fall into 51 categories. And Fig. 8 shows examples of point cloud models of 10 different categories from this dataset. It is especially important to note that these models are partial object rather than complete one. 400 point cloud models and 200 models from 10 categories of dataset are selected as the training set and testing set [5] (the views of same models in training dataset are different from that in testing dataset), respectively. In addition, 100 point cloud models of each object of the 10 categories are chosen.

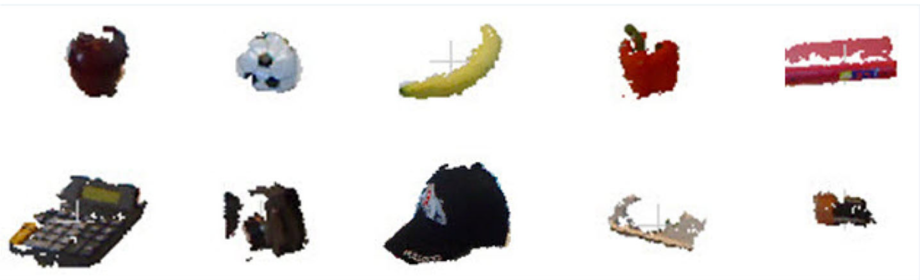


Fig. 8 Examples of point cloud models. Top row, left to right: apple, ball, banana, pepper, binder. Bottom row, left to right: calculator, camera, cap, cellphone, battery

5.2 Implementation details

The recognition pipeline shown in Fig. 1 are used to guide experiments. As for learning strategy, the classic and famous supervised learning algorithm, named Support Vector Machine (SVM), is adopted. And in the case of local-based process, Harris3D is employed as the major keypoint detection strategy for most of our experiments. Furthermore, all experiments were carried out on a computer with Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz and 16GB memory.

5.2.1 Selected filters

To demonstrate the effect of filters on the performance of object recognition, four more state-of-the-art filters are chosen for comparison, including normal-based Bilateral Filter (NBF), Moving Least Square (MLS) [4], Weighted Locally Projection (WLOP) [17] and Edge Aware Resample (EAR) [16].

5.2.2 Selected descriptors

Here, six different descriptors (shown in Table 1) are selected, including Spin Image (SI) [18], Signature of Histogram of Orientation (SHOT) [27], Fast Point Feature Histogram (FPFH) [26], Viewpoint Feature Histogram (VFH) [24], Cluttered Viewpoint Feature Histogram (CVFH) [3], Oriented, Unique and Repeatable CVFH (OURCVFH) [2], to conduct comparison.

The reason for choosing these filters and descriptors aforementioned is that they have a relatively high citation rate and are commonly used as comparable approaches with state-of-the-art performance. All methods, together with our proposed ones, are implemented using C++ and Point Cloud Library (PCL version 1.8).

5.2.3 Parameters

Particularly, it is necessary to notice that default parameters from the original papers or PCL implementations are used for the selected filters and descriptors.

For Guided 3D Point Cloud Filtering, the parameters ε and K or r should be specified. And the Iterative Guidance Normal Filter requires K of KNN, the number of normal filtering iterations n_{iter}, the standard variance parameter σ_s for spatial kernel, $\sigma_r$relating to signal weighting term, and the number of point updating p_{iter} to be adjusted. The selection of optimal values and effect of these parameters on performance are discussed in [12] and [13]

Table 1 Recognition accuracy obtained by our proposed descriptors with different radius

No.	Descriptors	Type	Length
1	SI	Local	225
2	SHOT	Local	352
3	FPFH	Local	33
4	VFH	Global	08
5	CVFH	Global	308
6	OURCVFH	Global	308

With respect to Local-to-Global Histogram, r is the key parameter, which will be discussed in the following section.

5.3 Descriptiveness

Two metrics are utilized to test the descriptiveness of proposed descriptors and demonstrate the recognition results of combination of different descriptors with different filters. The first metric is recognition accuracy which is evaluated as the ratio of correctly identified objects in the testing dataset. The second one is the confusion matrix, a well-suited tool of visualization of recognition performance on all 10 categories, with row representing the predicted classes while column indicating the actual classes. Given 100 models of each object of 10 categories, the prediction distribution percentage are calculated. Then filling out this specific table forms the confusion matrix of size 10×10.

5.4 Robustness

The robustness of LGH descriptors are evaluated from the perspective of different level of Gaussian noise and support radius.

5.4.1 Support radius

The setting of support radius has great influence on descriptiveness, robustness and efficiency of descriptors. Therefore, different support radius are adopted to determine the neighborhood of each keypoint which constitute the sub point cloud. The performance of LVFH and LESF with respect to different support radius are shown in Table 2 and corresponding confusion matrices are shown in Fig. 9.

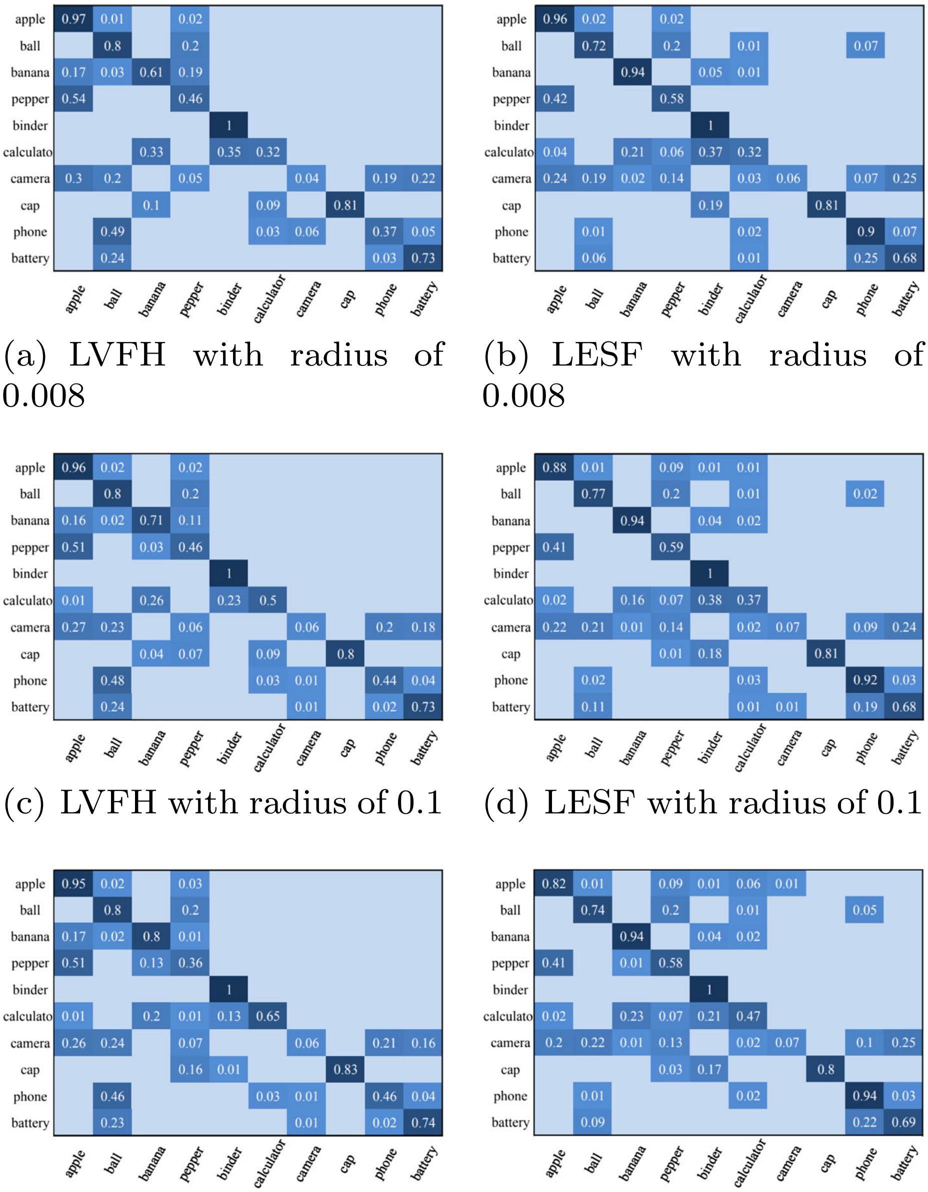
Five support radii are selected to carry out experiments. The following conclusions can be reached: First, the overall trend of the performance of proposed descriptors is incremental when the support radius increasing. This implies that large support radius determines a sub point cloud encapsulating more geometrical information and the corresponding descriptors have much higher descriptiveness. Second, LVFH and LESF produce the best performance with radius taking values of 0.05 and 0.15, respectively. Third, it can be reported from confusion matrices that LESF is able to attain much better performance on some object classes (such as banana and cellphone) where LVFH is not.

5.4.2 Gaussian noise

To evaluate the robustness of proposed features to noise, three different levels of Gaussian noise with standard deviations of 0.0005,0.0008 and 0.001 are added to each test point cloud, respectively. The experimental results are demonstrated in the first row of Tables 4, 5 and 6. The following summarisations can be got: First, the robustness of all descriptors tends to descend rapidly as the noise level increases. Second, the proposed LVFH and LESF

Table 2 Recognition accuracy obtained by our proposed descriptors with different radius

Radius	0.03	0.05	0.08	0.1	0.15
LVFH	58.5%	75.0%	73.0%	72.5%	74.5%
LESF	40.5%	69.5%	73.0%	73.5%	75.0%



(a) LVFH with radius of 0.008

(b) LESF with radius of 0.008

(c) LVFH with radius of 0.1

(d) LESF with radius of 0.1

(e) LVFH with radius of 0.15

(f) LVFH with radius of 0.15

Fig. 9 Confusion matrices on ten object classes of LVFH and LESF using different support radii

descriptors achieve the best performance in most cases, closely followed by FPFH. Specifically, under the noise with standard deviation of 0.0005, FPFH is slightly better than LVFH from the recognition accuracy view of point, while LESF simply gives a ranking fourth

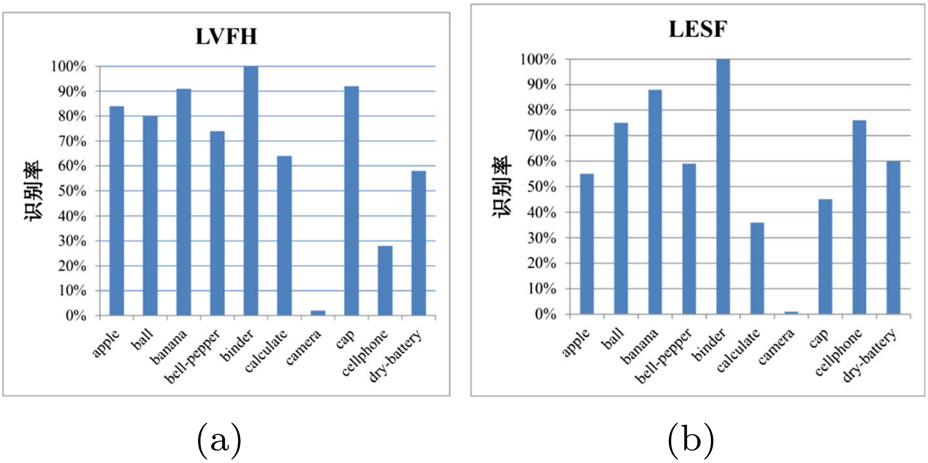


Fig. 10 Recognition accuracy of LVFH and LESF on each type of object

result. In the case of noise with standard deviation of 0.0008, LVFH becomes the descriptor with the best robustness, outperforming the other descriptors. And LESF comes second with a relatively high accuracy. As the standard deviation of noise increased to 0.001, LVFH obtains a considerably close performance compared to FPFH. However, LESF only achieves a moderate robustness with a medium accuracy result. It can be, therefore, inferred that LVFH is greatly robust to noise while LESF is less well-behaved dealing with noisy data. On the other hand, it is also indicated that noise reduction operations are completely necessary in the pipeline of 3D object recognition. And the influence of filters on the recognition performance will be discussed in later sections.

Figure 10 is an example of the histogram of recognition accuracy of LVFH and LESF on each of ten object classes under the noise with standard deviation of 0.0008. It can be reported that LVFH and LESF can achieve an acceptable result on most classes.

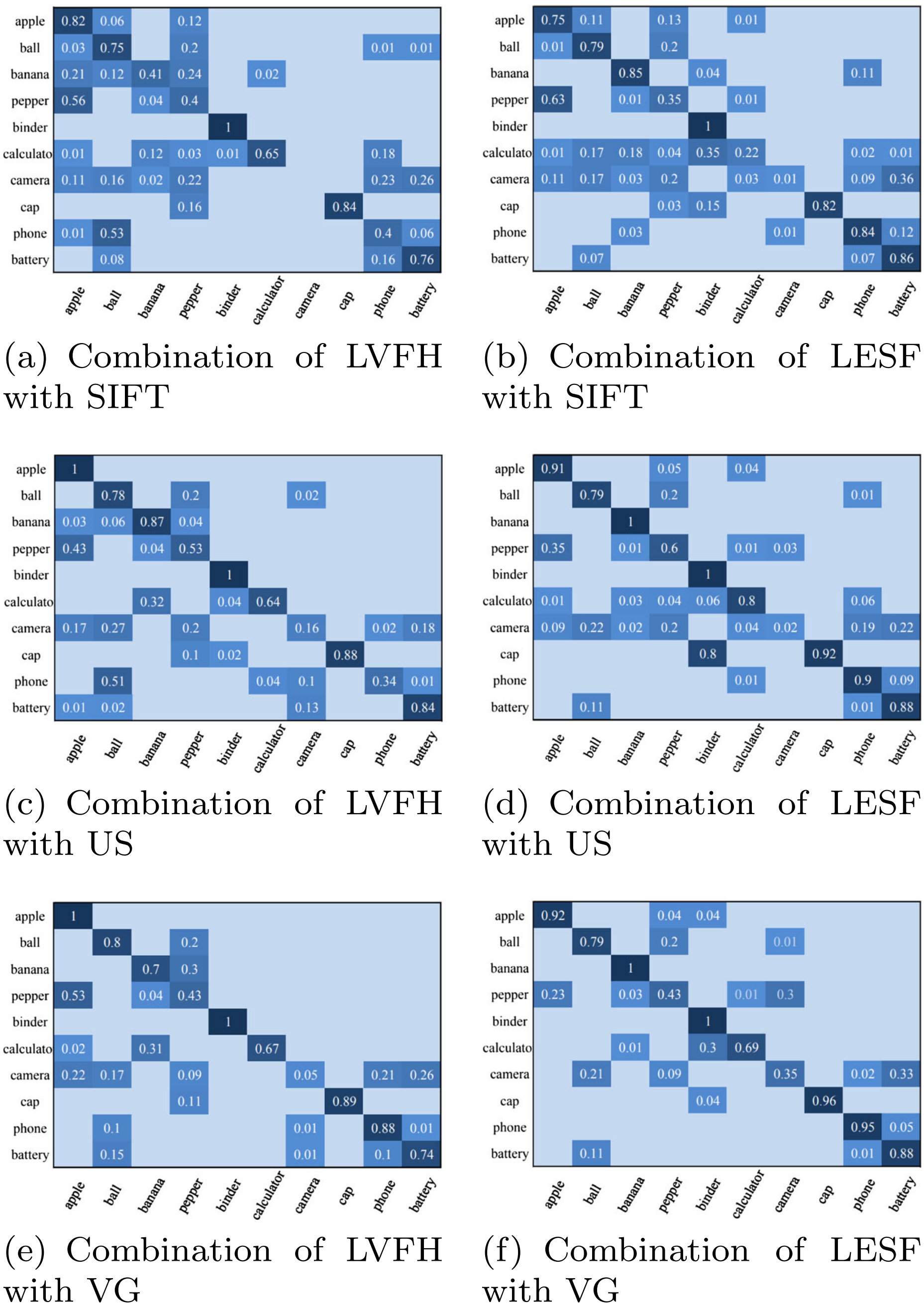
5.5 Combination with 3D keypoint detectors

Actually, the selection of keypoints plays an important role in feature extraction. Hence in order to demonstrate the effect of different keypoint detectors on the performance of proposed descriptors, Scale Invariant Feature Transform (extension of 2D SIFT to 3D point cloud), Harris3D, Uniform Sampling (US) and Voxel Grid (VG) algorithms are chosen. The overall performance of each detector-descriptor combination is presented in Table 3 and confusion matrices are shown in Fig. 11. Several major observations can be made as follows.

First, the overall performance of combination of proposed descriptors with Uniform Sampling is much better than other combinations.

Table 3 Recognition accuracy obtained by our proposed descriptors with different 3D keypoint detectors

Detectors	SIFT	HARRIS3D	US	VG
LVFH	73.0%	74.5%	80.0%	76.5%
LESF	78.0%	75.0%	**81.0%**	**85.5%**



(a) Combination of LVFH with SIFT

(b) Combination of LESF with SIFT

(c) Combination of LVFH with US

(d) Combination of LESF with US

(e) Combination of LVFH with VG

(f) Combination of LESF with VG

Fig. 11 Confusion matrices on ten object classes of combinations of LVFH and LESF with SIFT, Uniform Sampling and Voxel Grid

Table 4 Recognition accuracy obtained by combination of different filters with different descriptors tested on dataset with Gaussion noise(σ=0.0005)

Filters	SI	SHOT	FPFH	VFH	CVFH	OURCVFH	LVFH	LESF
-	70.0%	64.5%	73.5%	56.0%	53.5%	46.5%	71.0%	68.5%
NBF	68.5%	65.0%	69.0%	50.0%	53.5%	44.0%	70.5%	69.0%
MLS	32.5%	48.0%	42.0%	32.5%	61.0%	49.0%	44.0%	34.5%
WLOP	55.0%	58.0%	63.5%	61.5%	63.0%	54.5%	60.5%	62.0%
EAR	60.5%	65.0%	68.0%	66.5%	66.0%	52.5%	69.0%	63.5%
G3DF	62.0%	66.0%	71.5%	67.5%	64.5%	53.0%	71.5%	68.5%
IGNF	67.0%	74.0%	72.0%	48.5%	61.5%	55.5%	72.5%	66.0%

Second, specifically, the combination of LVFH with Uniform Sampling and combination of LESF with Voxel Grid achieve the best performance in their respective comparison experiments. And the overall performance of LVFH and LESF are almost the same when combined with SIFT and Harris3D algorithms.

Third, from the confusion matrices, it can be stated that all these combinations can get acceptable recognition accuracy on most of ten classes. In particular, the overall performance of LESF combining with Uniform Sampling and Voxel Grid is a little higher than the combination of LVFH with these two detectors.

Fourth, these experimental results also reveal the fact that the selection of suitable 3D keypoint detectors can boost the performance of object recognition.

6 Noisy 3D point cloud based object recognition

Up to now, the G3DF, IGNF and the Local-to-Global Histogram descriptors have been introduced. In order to further demonstrate the functionalities, properties and performance of these proposed filters and descriptors, the complete noisy 3D point cloud based object recognition pipeline proposed in Section 2 is used to carry out the experiments.

G3DF and IGNF are used together with the other four filters as pre-processing algorithms. And LGH descriptors with the other six descriptors are integrated as feature extraction algorithms. These different combinations of filters and descriptors are evaluated

Table 5 Recognition accuracy obtained by combination of different filters with different descriptors tested on dataset with Gaussion noise(σ=0.0008)

Filters	SI	SHOT	FPFH	VFH	CVFH	OURCVFH	LVFH	LESF
-	61.5%	59.5%	61.5%	33.5%	52.5%	44.0%	64.5%	62.0%
NBF	65.5%	65.0%	63.0%	32.5%	10.05%	45.5%	61.5%	56.0%
MLS	37.0%	50.0%	37.5%	33.0%	60.5%	49.5%	44.0%	37.0%
WLOP	58.0%	60.0%	63.0%	60.5%	62.0%	54.5%	63.5%	23.0%
EAR	62.0%	69.5%	74.5%	70.0%	64.0%	52.0%	74.5%	72.0%
G3DF	64.0%	69.5%	69.5%	68.5%	64.0%	52.5%	71.0%	72.0%
IGNF	65.5%	73.5%	72.5%	50.0%	61.5%	58.0%	75.0%	72.0%

Table 6 Recognition accuracy obtained by combination of different filters with different descriptors tested on dataset with Gaussion noise(σ=0.001)

Filters	SI	SHOT	FPFH	VFH	CVFH	OURCVFH	LVFH	LESF
-	53.0%	54.5%	55.5%	14.0%	29.0%	39.0%	54.5%	47.0%
NBF	57.0%	60.0%	55.5%	13.5%	28.0%	38.0%	53.5%	48.0%
MLS	33.5%	49.0%	41.0%	33.0%	60.5%	48.5%	42.0%	34.0%
WLOP	58.0%	62.0%	64.5%	59.5%	61.0%	53.0%	65.5%	25.0%
EAR	63.0%	71.0%	73.5%	63.5%	61.5%	52.5%	73.5%	72.0%
G3DF	65.5%	68.5%	73.5%	64.0%	61.5%	53.5%	74.5%	70.5%
IGNF	69.5%	73.0%	74.0%	50.0%	61.5%	58.0%	76.0%	75.5%

on the testing date with three different levels of Gaussian noise mentioned in the previous section.

6.1 Performance and Discussion

The recognition results of these different combinations under different levels of Gaussian noise are shown in Tables 4, 5 and 6. Figure 12 gives the corresponding histogram representation. Specifically, Figs. 13 and 14 show the confusion matrices of combinations of G3DF, IGNF with LVFH, LESF on each object of ten classes, respectively. Several major observations can be made from these results.

First, generally, the incorporation of pre-processing operations (filtering) can help improve the quality and accuracy of 3D point cloud, which contributes to the amelioration of descriptiveness. The overall performance of object recognition produced by combinations of filters and descriptors is much higher than these without filters, although these exists some exceptions, such as combinations of MLS with SI, IGNF with LESF in Table 4. In particular, it is worth noting that as the noise level increases, the margins of the improvement of recognition accuracy brought by these combinations become much more significant.

Second, it is clear that in most cases, LVFH and LESF generally show a much better performance compared to other descriptors when combined with different filters. While

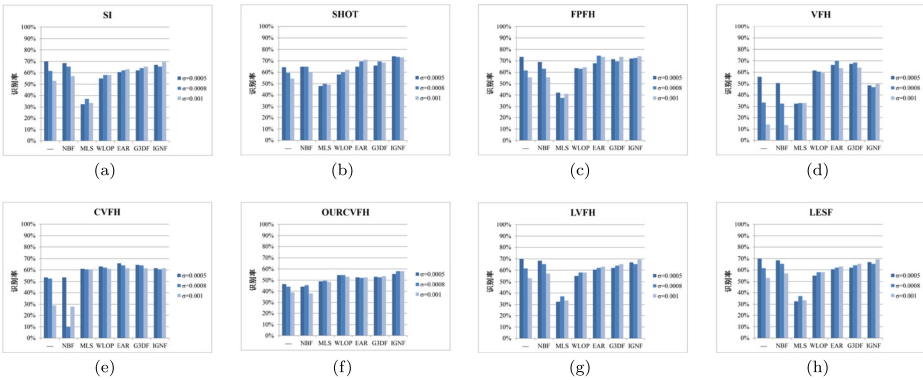
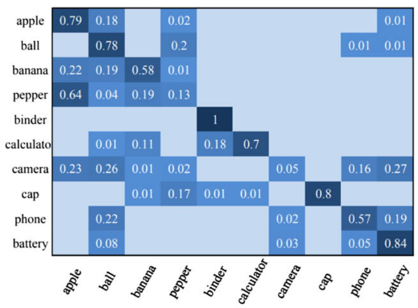
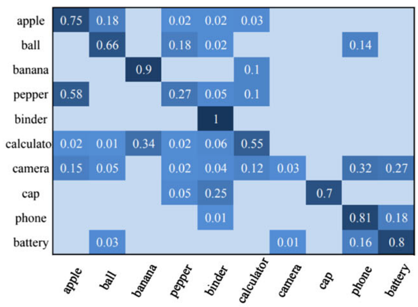


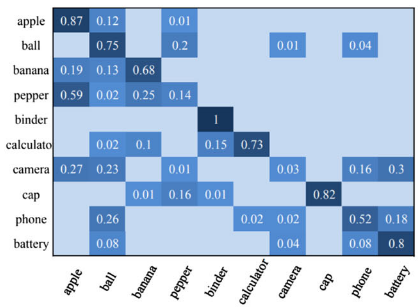
Fig. 12 Histograms of combinations of different filters with different feature descriptors in the cases of three different level of noise



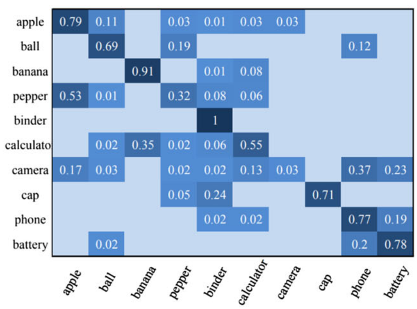
(a) Combination of G3DF with LVFH under noise with standard deviation of 0.0005



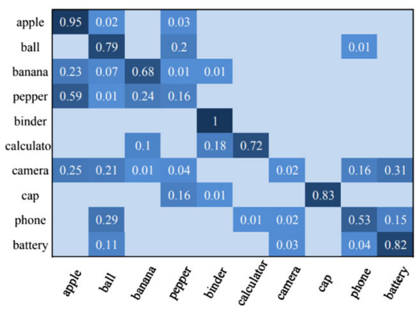
(b) Combination of G3DF with LESF under noise with standard deviation of 0.0005



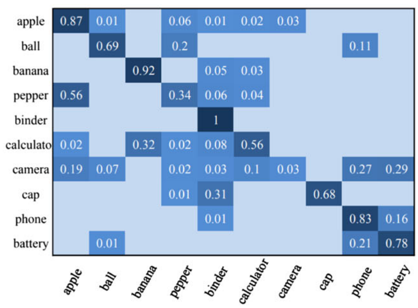
(c) Combination of G3DF with LVFH under noise with standard deviation of 0.0008



(d) Combination of G3DF with LESF under noise with standard deviation of 0.0008

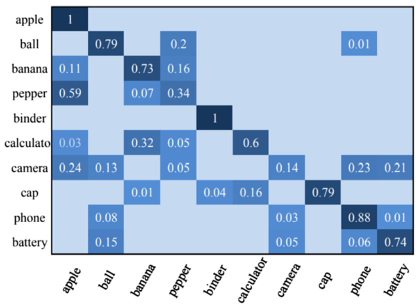


(e) Combination of G3DF with LVFH under noise with standard deviation of 0.001

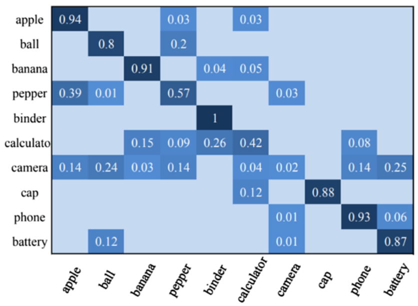


(f) Combination of G3DF with LESF under noise with standard deviation of 0.001

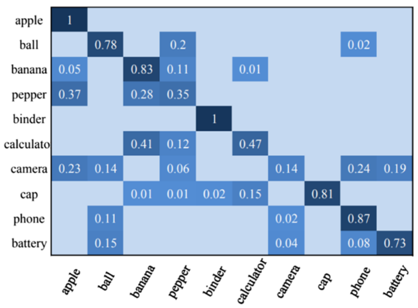
Fig. 13 Confusion Matrices of combination of our G3DF with LVFH and LESF in the presence of noise



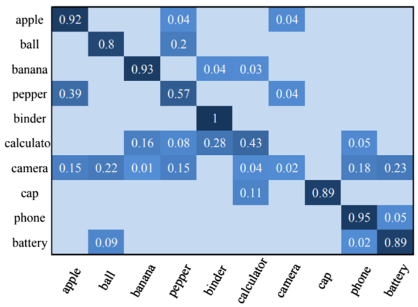
(a) Combination of IGNF with LVFH under noise with standard deviation of 0.0005



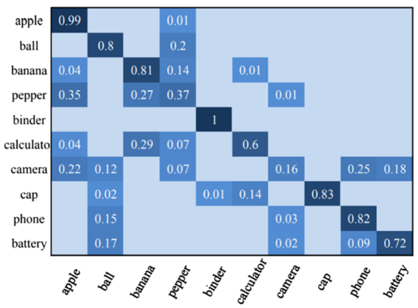
(b) Combination of IGNF with LESF under noise with standard deviation of 0.0005



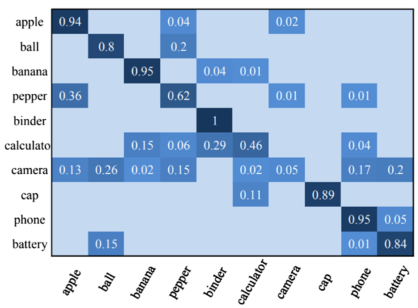
(c) Combination of IGNF with LVFH under noise with standard deviation of 0.0008



(d) Combination of IGNF with LESF under noise with standard deviation of 0.0005



(e) Combination of IGNF with LVFH under noise with standard deviation of 0.001



(f) Combination of IGNF with LESF under noise with standard deviation of 0.001

Fig. 14 Confusion Matrices of combination of our IGNF with LVFH and LESF in the presence of noise

G3DF and IGNF are the top filters outperforming other filters in terms of combination with different descriptors.

Third, specifically, under low-level Gaussian noise with standard deviation of 0.0005, the assembly of LVFH with IGNF acquires the best performance in terms of recognition accuracy, followed by that with G3DF. And the performance of combinations of LESF with G3DF and IGNF comes with the second and third places under the same conditions, respectively. As the standard deviation increases to 0.0008, the integration of LVFH with IGNF behaves best, while LESF gives the highest values of recognition accuracy when combined with G3DF and IGNF. It can be further concluded that in the case of models with low-level or medium-level noise, G3DF produces almost the same response as IGNF from the perspective of improving the representation ability of LVFH and LESF. Under noise with standard deviation of 0.001, both LVFH and LESF obtain the top performance when combined with IGNF. And they come in the ranking second and third places respectively, when combined with G3DF. It can be indicated that with respect to high-level noise, IGNF performs much better than G3DF for improvement of descriptiveness of LVFH and LESF.

Fourth, it can be further found from the confusion matrices that four combinations G3DF with LVFH, G3DF with LESF, IGNF with LVFH and IGNF with LESF are able to yield satisfyingly acceptable recognition result on most of objects of ten classes. It can be further concluded that G3DF and IGNF actually play a positive role in helping improve the expression ability of descriptors, especially LVFH and LESF. And these four combinations become combinations with high-level performance, which outperforms the other cases.

6.2 Efficiency

Computational efficiency is another important metric for noisy 3D object recognition performance. Therefore, in order to give a thorough evaluation, the average time will be evaluated on testing dataset with three different level of noise by using the aforementioned combinations for object recognition.

It can be concluded from Table 7 that 1) under the same conditions, G3DF combining with different descriptors computationally outperforms other groups by a large margin, while IGNF is relatively time-consuming, only faster than EAR. 2) In the context of object recognition, the combination of LVFH with G3DF provides a trade-off between efficiency and recognition accuracy. Therefore, it is suitable for real-time applications. 3) On the other hand, although the combinations of LESF with G3DF and IGNF, LVFH with IGNF run a little slower, they achieve the best recognition performance. Consequently, they can be applied to applications that require high-quality or high-accuracy point cloud models.

Table 7 Average computational time by using combination of different filters with different descriptors

Filters	SI	SHOT	FPFH	VFH	CVFH	OURCVFH	LVFH	LESF
NBF	525.71	1,243.38	1,009.39	392.58	1,361.98	1,401.18	799.01	1,500.26
MLS	473.22	1,191.20	965.22	336.60	1,316.61	1,340.82	973.44	1,396.75
WLOP	833.49	1,550.18	1,310.42	698.23	1,671.55	1,714.30	1,115.98	1,812.21
EAR	2,525.14	3,240.44	3,014.64	2,390.80	3,360.72	3,487.30	2,894.50	3,391.49
G3DF	267.59	984.36	759.55	135.64	1,143.16	1,150.96	503.36	1,022.47
IGNF	1,077.24	1,796.80	1,572.93	944.12	1,954.60	1,964.48	1,301.71	2,079.14

7 Conclusion

In this paper, the filtering operation is assembled into the 3D point cloud based object recognition framework. To demonstrate the effectiveness of this pipeline, an overview of proposed filters: Guided 3D Point Cloud Filter and Iterative Guidance Normal Filter were first outlined. Then, a new kind of descriptor, named Local-to-Global Histogram was proposed, which includes Local Viewpoint Feature Histogram and Local Ensemble of Shape Function. Experimental results showed the effectiveness of proposed descriptors in terms of descriptiveness, robustness and combination with different keypoint detectors. Finally, many filters and descriptors combinations were integrated into this framework to conduct experiments which stated the feasibility of designed framework and further verified the performance of these proposed algorithms. In the future, deep learning strategies will be taken into account in the designed recognition framework to use its powerful learning ability.

Acknowledgements This research was supported by the National Natural Science Foundation of China (No. 62002299), and the Natural Science Foundation of Chongqing of China (No. cstc2020jcyj-msxmX0126), and the Fundamental Research Funds for the Central Universities (No. SWU120005)

References

1. Aldoma A, Marton ZC, Tombari F, Wohlkinger W, Potthast C, Zeisl B, Rusu RB, Gedikli S, Vincze M (2012) Tutorial: Point cloud library: Three-dimensional object recognition and 6 dof pose estimation. Robotics and Automation Magazine 19(3):80–91
2. Aldoma A, Tombari F, Rusu RB, Vincze M (2012) Our-cvfh - oriented, unique and repeatable clustered viewpoint feature histogram for object recognition and 6dof pose estimation. In: Joint DAGM (German Association for Pattern Recognition) and OAGM symposium
3. Aldoma A, Vincze M, Blodow N, Gossow D, Gedikli S, Rusu RB, Bradski GR (2011) Cad-model recognition and 6dof pose estimation using 3d cues. In: IEEE International conference on computer vision workshops
4. Alexa M, Behr J, Cohen-Or D, Fleishman S, Levin D, Silva CT (2015) Computing and rendering point set surfaces. IEEE Transactions on Visualization & Computer Graphics 9(1):3–15
5. Alexandre LA (2012) 3d descriptors for object and category recognition: A comparative evaluation. In: IEEE International conference on intelligent robotic systems workshops
6. Behley J, Steinhage V, Cremers AB (2012) Performance of histogram descriptors for the classification of 3d laser range data in urban environments. In: IEEE International conference on robotics & automation
7. Chen X, Kundu K, Zhang Z, Ma H, Fidler S, Urtasun R (2016) Monocular 3d object detection for autonomous driving. In: IEEE Conference on computer vision and pattern recognition
8. Creusot C, Pears N, Austin J (2013) A machine-learning approach to keypoint detection and landmarking on 3d meshes. Int J Comput Vis 102(1-3):146–179
9. de Oliveira Rente P, Brites C, Ascenso J, Pereira F (2019) Graph-based static 3d point clouds geometry coding. IEEE Transactions on Multimedia 21(2):284–299. https://doi.org/10.1109/TMM.2018.2859591
10. Guo Y, Sohel F, Bennamoun M, Lu M, Wan J (2013) Rotational projection statistics for 3d local surface description and object recognition. Int J Comput Vis 105(1):63–86
11. Hadji I, Desouza GN (2014) Local-to-global signature descriptor for 3d object recognition. In: Asian conference on computer vision
12. Han XF, Jin JS, Wang MJ, Jiang W (2017) Guided 3d point cloud filtering. Multi Tools App 77(3):1–15
13. Han XF, Jin JS, Wang MJ, Jiang W (2017) Iterative guidance normal filter for point cloud. Multi Tools App 77(1):1–16
14. Han XF, Jin JS, Wang MJ, Jiang W, Gao L, Xiao L (2017) A review of algorithms for filtering the 3d point cloud. Signal Processing Image Communication 57:103–112
15. He K, Sun J, Tang X (2013) Guided image filtering. IEEE Transactions on Pattern Analysis & Machine Intelligence 35(6):1397–1409
16. Huang H, Wu S, Gong M, Cohen-Or D, Zhang H (2013) Edge-aware point set resampling. Acm Transactions on Graphics 32(1):1–12

17. Hui H, Dan L, Hao Z, Ascher U, Cohen-Or D (2009) Consolidation of unorganized point clouds for surface reconstruction. In: Acm Siggraph Asia
18. Johnson AE, Hebert M (1998) Surface matching for object recognition in complex three-dimensional scenes. Image & Vision Computing 16(9-10):635–651
19. Kasaei SH, Tomé AM, Lopes LS, Oliveira M (2016) Good: A global orthographic object descriptor for 3d object recognition and manipulation. Pattern Recogn Lett 83:312–320
20. Lai K, Bo L, Ren X, Fox D (2011) A large-scale hierarchical multi-view rgb-d object dataset. In: IEEE International conference on robotics & automation
21. Ma C, Guo Y, Yang J, An W (2018) Learning multi-view representation with lstm for 3d shape recognition and retrieval. IEEE Transactions on Multimedia, p 1–1. https://doi.org/10.1109/TMM.2018.2875512
22. Marc K, Qian-Yi Z, Vladlen K (2017) Learning compact geometric features. In: IEEE International conference on computer vision
23. Qi Z, Shen X, Li X, Jia J (2014) Rolling guidance filter. In: European conference on computer vision
24. Rusu RB, Bradski G, Thibaux R, Hsu J (2014) Fast 3d recognition and pose using the viewpoint feature histogram. In: IEEE/RSJ International conference on intelligent robots & systems
25. Rusu RB (2009) Semantic 3d object maps for everyday manipulation in human living environments. Ph.D. Thesis, Computer Science department, Technische Universitaet Muenchen, Germany
26. Rusu RB, Blodow N, Beetz M (2009) Fast point feature histograms (fpfh) for 3d registration. In: IEEE International conference on robotics & automation
27. Salti S, Tombari F, Di Stefano L (2014) Shot: Unique signatures of histograms for surface and texture description. Computer Vision & Image Understanding 125(8):251–264
28. Sun L, Zhao C, Yan Z, Liu P, Duckett T, Stolkin R (2019) A novel weakly-supervised approach for rgb-d-based nuclear waste object detection. IEEE Sensors J 19(9):3487–3500
29. Sun X, Rosin P, Martin R, Langbein F (2007) Fast and effective feature-preserving mesh denoising. IEEE Trans Vis Comput Graph 13(5):925–938
30. Tabia H, Laga H (2015) Covariance-based descriptors for efficient 3d shape matching, retrieval, and classification. IEEE Transactions on Multimedia 17(9):1591–1603. https://doi.org/10.1109/TMM.2015.2457676
31. Tombari F, Salti S, Stefano LD (2013) Performance evaluation of 3d keypoint detectors. Int J Comput Vis 102(1-3):198–220
32. Wang PS, Fu XM, Liu Y, Tong X, Liu SL, Guo B (2015) Rolling guidance normal filter for geometric processing. Acm Transactions on Graphics 34(6):1–9
33. Wu P, Liu Y, Ye M, Li J, Du S (2017) Fast and adaptive 3d reconstruction with extensively high completeness. IEEE Transactions on Multimedia 19(2):266–278. https://doi.org/10.1109/TMM.2016.2612761
34. Xian-Feng H, Shi-Jie S, Xiang-Yu S, Xiao G-Q (2020) 3d point cloud descriptors in hand-crafted and deep learning age: State-of-the-art. arXiv:1802.02297v2
35. Yulan G, Mohammed B, Ferdous S (2014) 3d object recognition in cluttered scenes with local surface features: A survey. IEEE Transactions on Pattern Analysis & Machine Intelligence 36(11):2270–87
36. Zhao H, Tang M, Ding H (2020) Hoppf: A novel local surface descriptor for 3d object recognition. Pattern Recogn 103:107272
37. Zhou Y, Tuzel O (2018) Voxelnet: End-to-end learning for point cloud based 3d object detection. In: IEEE Conference on computer vision and pattern recognition