International Journal of Computer Vision
https://doi.org/10.1007/s11263-025-02425-3

A Fast and Lightweight 3D Keypoint Detector

Chengzhuan Yang[1] · Qian Yu[2] · Hui Wei[3] · Fei Wu[4] · Yunliang Jiang[1] · Zhonglong Zheng[1] · Ming-Hsuan Yang[5]

Received: 29 August 2024 / Accepted: 1 March 2025

Abstract
Keypoint detection is crucial in many visual tasks, such as object recognition, shape retrieval, and 3D reconstruction, as labeling point data is labor-intensive or sometimes implausible. Nevertheless, it is challenging to quickly and accurately locate keypoints unsupervised from point clouds. This work proposes a fast and lightweight 3D keypoint detector that can efficiently and accurately detect keypoints from point clouds. Our method does not require a complex model learning process and generalizes well to new scenes. Specifically, we consider detecting keypoints a saliency detection problem for a point cloud. First, we propose a simple and effective distance measure to characterize the saliency of points in a point cloud. This distance describes geometrically essential points in the point cloud. Next, we present a regional saliency based on relative centroid distance representation that can globally characterize keypoints with regional visual information. Third, we combine geometric and semantic cues to generate a saliency map of the point cloud for determining stable 3D keypoints. We evaluate our method against existing approaches on four benchmark keypoint datasets to demonstrate its state-of-the-art performance.

Keywords 3D keypoint detection · Point cloud · Point saliency · Saliency detection

Communicated by Kwang Moo Yi.

✉ Zhonglong Zheng
zhonglong@zjnu.edu.cn

✉ Ming-Hsuan Yang
mhyang@ucmerced.edu

Chengzhuan Yang
czyang@zjnu.edu.cn

Qian Yu
yuqian@jsut.edu.cn

Hui Wei
weihui@fudan.edu.cn

Fei Wu
wufei@zju.edu.cn

Yunliang Jiang
jyl@zjhu.edu.cn

[1] School of Computer Science and Technology, Zhejiang Normal University, Jinhua, China

[2] School of Computer Engineering, Jiangsu University of Technology, Changzhou, China

[3] Laboratory of Cognitive Algorithm and Model, School of Computer Science, Fudan University, Shanghai, China

[4] School of Computer Science and Technology, Zhejiang University, Hangzhou, China

[5] University of California at Merced, Merced, CA, USA

1 Introduction

Keypoint detection is an integral part of many tasks in computer vision and robotics, such as SLAM based on point cloud (Boroson & Ayanian, 2019; Hu et al., 2024; Jelavic et al., 2022), pose estimation (Deng et al., 2022; Geng et al., 2021; Zhang et al., 2021), 3D object recognition (Uy & Lee, 2018; Yang et al., 2017), and shape registration (Shi et al., 2021; Wang et al., 2018, 2022). It plays a significant role in performing these visual tasks by quickly and robustly extracting geometrically and visually consistent keypoints from point clouds, as incorrect detection can negatively affect these tasks. Given the importance of this problem, keypoint detection has attracted significant interest recently (Bai et al., 2023; Barroso-Laguna & Mikolajczyk, 2022; Gao et al., 2023; Lu & Koniusz, 2022; Lu et al., 2020; Luo et al., 2022; Yang & Pavone, 2023; Zhang et al., 2024; Zheng et al., 2022; Zohaib & Del Bue, 2023).

Existing handcrafted methods typically detect 3D keypoints from point clouds using local geometric statistics such as mesh saliency (Lee et al., 2005), ISS (Zhong, 2009), Harris-3D (Sipiran & Bustos, 2011), and SIFT-3D (Rister et al., 2017). These methods only consider simple geometric features and do not use global semantic information, which often leads to the instability of the detected key-

Published online: 01 April 2025

points in the presence of noise and density changes in the point cloud. Recently, several learning-based keypoint detectors have been proposed, including USIP (Li & Lee, 2019), D3Feat (Bai et al., 2020), UKPAGN (You et al., 2022), and SNAKE (Zhong et al., 2022). These learning-based keypoint detectors improve the performance of keypoint detection by utilizing a large amount of point cloud data to train the model. However, these methods typically require complex training processes, and the model size is often large. Since these methods consider only the object's global semantic cues, they do not effectively fuse important geometric structure information, affecting detection accuracy.

This work proposes a fast and lightweight 3D keypoint detector (FL3K) to address these issues. We consider keypoint detection a saliency detection problem for point clouds. Its saliency value determines whether a point is a keypoint. First, we present a simple, effective relative center distance measure called geometric saliency to capture the structural information of a point cloud. Second, we identify the points with high saliency values based on their relative center distance to characterize the global semantic information. Finally, we use a nonlinear suppression aggregation method to effectively combine geometric and regional saliency measures to determine the stable keypoints. Extensive experimental results on four benchmark datasets demonstrate that our method achieves state-of-the-art performance, outperforming existing handcrafted and learning-based approaches. Figure 1 shows examples of qualitative keypoint detection using our method. The model accurately detects 3D keypoints on rigid and nonrigid 3D objects, consistent with human visual perception. In addition, the detected keypoints are visually salient with clear semantics, such as the tail of an airplane, the four legs of a chair, and the hands and feet of a human.

The main contributions of this work are:

- We propose a fast, lightweight 3D keypoint detector framework based on point saliency.
- We define two measures of point clouds: geometric and regional saliency.
  These two measures can effectively describe a point cloud's geometric structural and semantic information.
- We propose a weighted nonlinear aggregation method to effectively integrate geometric and regional saliency cues.
- Our method achieves state-of-the-art detection, surpassing the performance of existing handcrafted and learning-based keypoint detectors.

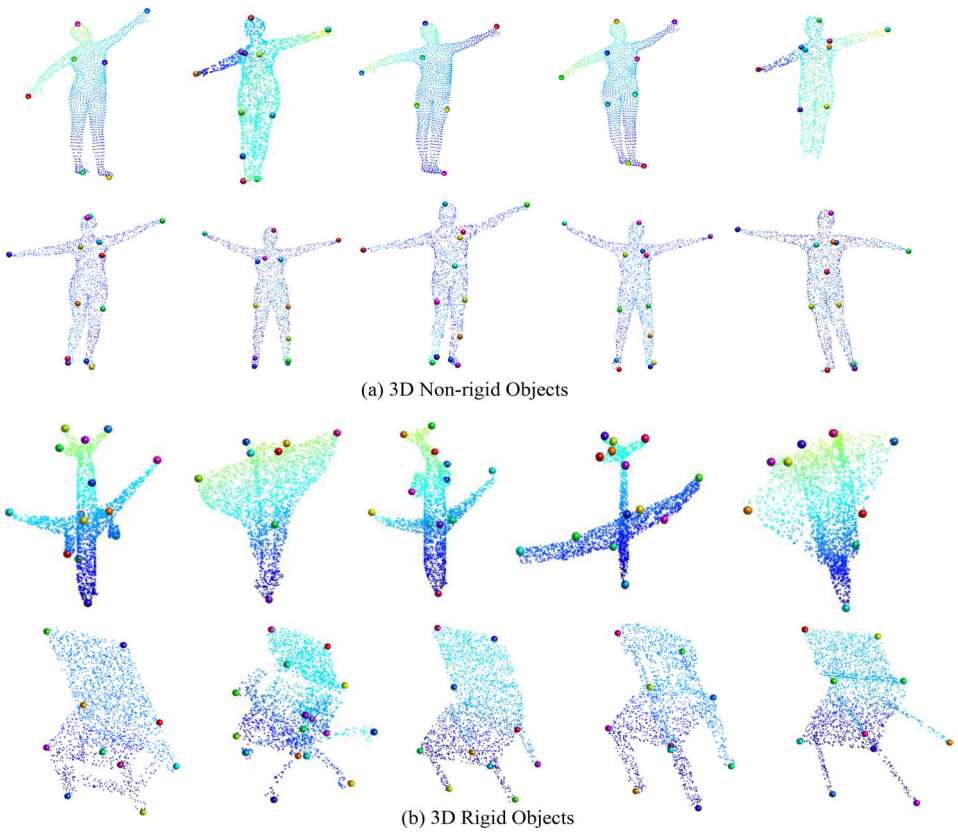


Fig. 1 Detected keypoints on rigid and nonrigid 3D objects using the proposed FL3K method. The keypoints detected on 3D rigid and non-rigid objects are accurate and consistent with human visual perception. Meanwhile, these keypoints are visually salient and semantically meaningful, such as the tail of an airplane, the chair legs, and human hands as well as feet

2 Related Work

2.1 Handcrafted Methods

Tombari et al. (2013) categorize 3D keypoint detection approaches into fixed-scale and adaptive-scale detectors. The size of local support is an input parameter for fixed-scale methods, and representative schemes include local surface patches (LPS) (Chen & Bhanu, 2007), ISS (Zhong, 2009), and HKS (Sun et al., 2009). LPS operates on the maximum and minimum principal curvatures of a point cloud. A point is considered a keypoint if it is a global extremum in a predefined neighborhood. On the other hand, ISS selects salient points with local neighborhoods with large variations along each principal axis. HKS serves as a saliency measure by restricting the heat kernel diffusion process on a mesh.

Adaptive-scale methods can determine the size of the local support using scale-space analysis. Representative methods include MeshDoG (Zaharescu et al., 2009), Laplace-Beltrami Scale Space (LBSS) (Unnikrishnan & Hebert, 2008), Salient Points (SP) (Castellani et al., 2008), Harris-3D (Sipiran & Bustos, 2011), and SIFT-3D (Rister et al., 2017). MeshDoG and SP construct the scale space of curvature by using Gaussian difference operators and then selecting a point with local extrema near the ring as the keypoint. On the other hand, LBSS computes saliency by applying Laplace-Beltrami operators on supports around each point of a shape. Harris-3D extends the Harris corner detector to 3D meshes, and SIFT-3D extends the SITF algorithm to keypoint detection in three-dimensional images. While these methods can detect 3D keypoints well, all require a hand-crafted saliency function in the local neighborhood of each data point. As such, the performance depends on the effectiveness of the handcrafted saliency function. This function is susceptible to the complexity of the object, density variations, and noise, which leads to less stable results.

2.2 Learning-Based Methods

Learning-based approaches train classifiers from a large corpus of data for keypoint detection. For example, Yew and Lee (2018) propose a 3DFeat-Net model for point cloud registration, capable of learning keypoint detection and description of 3D object shapes from laser point clouds in a weakly supervised learning manner. S3DFeat-Net's trainingt-Net relies on learning distinguishable descriptors through a Siamese network with an attention score map that estimates the saliency of each point as its by-product, this method fails to ensure superior performance for keypoint detection. In Li and Lee (2019), present an unsupervised stable interest point (USIP) detection method from a 3D point cloud, which regresses keypoint locations from pre-segmented local groups and achieves good detection accuracy. Bai et al. (2020) develop a D3Feat method for jointly learning keypoint scores and local features from a point cloud. The method relies on an auxiliary task of correctly estimating rotations in the Siamese structure, ignoring the semantic information. You et al. (2022) introduce a generalized self-supervised 3D keypoint detector called UKPGAN. This method provides two GAN-based keypoint sparse control modules and salient information distillation to locate important keypoints.

Recently, Zhong et al. (2022) design a shape-aware neural 3D keypoint field to locate keypoints. This method achieves keypoint detection through shape reconstruction and achieves good detection performance. In Zohaib and Del Bue (2023), Zohaib et al. propose a self-supervised and coherent 3D keypoint detection method called SC3K, which is robust to the presence of point cloud rotations, noise, and density changes. A self-supervised 3D implicit Transporter method is developed by Zhong et al. (2023) to discover temporally correspondent keypoints from point cloud sequences. It adopts a closed-loop control strategy for object manipulation based on the detected keypoints. Wimmer et al. (2024) propose a few-shot 3D keypoint detection method based on back-projected 2D features. This method applies the features from a large pre-trained 2D vision model to 3D shapes for keypoint detection. Additionally, they use a keypoint candidate optimization module to improve the accuracy of keypoint detection. However, this method requires at least three labeled samples for model training. Further, this method can only handle polygonal 3D mesh data and cannot be directly applied to the analysis of point cloud data, thus limiting the scope of the technique. While these methods perform well in some keypoint detection tasks, they are less effective for point clouds with rigid and nonrigid variations. In addition, existing learning-based methods require complex training processes, and the models need to be more lightweight, limiting their applicability. More importantly, these schemes need to be generalized well in real-life situations.

Closely relevant to our work are point cloud saliency maps (PCSM) (Zheng et al., 2019) and saliency-based keypoint detectors (SKD) (Tinchev et al., 2021). The PCSM method assigns a score to each point in the point cloud to reflect its contribution to the model's recognition loss. The point cloud saliency scores are computed by performing the proposed point-dropping operations and using a differentiable procedure to move the point toward the center point of the cloud to approximate the point-dropping operation. However, this method mainly focuses on the performance of point cloud classification tasks and has yet to be applied to keypoint detection and registration tasks. In addition, training the network model requires the provision of category labels. The SKD method first takes the gradient information of a pre-trained neural network model as the initial saliency score. The saliency score is combined with the point cloud contex-

tual features and PCA features of point descriptors as input to train a multi-layer neural network model for keypoint detection. Both PCM and SKD methods require more complex model training processes. In contrast, our method performs keypoint detection by explicitly combining geometric and regional saliency. The proposed FL3K method can combine the geometric structural information of the point cloud with the semantic information, which achieves better results in the keypoint detection and point cloud registration tasks. In addition, our method is lightweight and efficient.

3 FL3K Detector

3.1 Overview

For a point cloud $X = \{x_i | x_i \in \mathbb{R}^3, i = 1, 2, \ldots, N\}$, where x denotes the 3D spatial coordinates of objects in a natural scene obtained from a 3D sensor. We aim to find a set of keypoints $\tilde{X} \subseteq X$ that is consistent with human visual perception regarding geometric and semantic information, where $|\tilde{X}|$ is the number of keypoints.

We propose the FL3K method, which detects keypoints by utilizing the saliency of points. Figure 2 illustrates the main modules of our method. First, we calculate the geometric saliency S_{geo} for an input point cloud X, as shown in Fig. 2b. This saliency reflects prominent geometric features in the point cloud, such as high curvature and corner points. Then, we calculate the regional saliency S_r, as shown in Fig. 2c. This saliency is mainly used to capture the distinctness of entire semantic parts for a point cloud. Third, we obtain the final saliency map S by fusing geometric and regional saliency, as shown in Fig. 2d. Finally, we generate keypoint results based on the saliency map S, as shown in Fig. 2e.

3.2 Geometric Saliency

When humans perceive 3D objects, points with important geometric characteristics usually carry more visual information than others. Figure 3a shows a 3D object and its ground-truth keypoints. Note that the ground-truth keypoints are concentrated at positions with important geometric features, such as an object's corners and edges. Assigning high importance to the points at these locations is important for keypoint detection. As such, we propose a relative center distance (RCD) representation to describe these geometrically significant points effectively.

For each point $p \in X$, we determine the set of spherical neighborhood points $N(p)$ with a radius r. In this work, $N(p)$ is defined as $N(p) = \{p_j | L_2(p_j, p) < r\}$, where L_2 is the Euclidean distance. We set $r = 15$, the mesh resolution (mr) in the experiment, where mr denotes the resolution of the point cloud. The mr can be obtained by calculating the average distance between the nearest point pairs in the point cloud. Then, we compute the geometric centers of all neighboring points

$$\mu_g = \frac{1}{|N(p)|} \sum_{|N(p)|} p_j, \tag{1}$$

where $|N(p)|$ is the number of neighbor points for p. Next, we compute the Euclidean distance $d(p)$ between point p and geometric center μ_g. Finally, we compute the RCD representation of the point p,

$$RCD(p) = \frac{d(p)}{r}, \tag{2}$$

where the RCD can effectively characterize geometrically important points of a 3D object. The larger the RCD of a point, the more significant the point is. Thus, we use $RCD(p)$ as the geometric saliency $S_{geo}(p)$ of point p. Figure 3b shows the RCD of an edge point, and Fig. 3c shows the RCD of an interior point. The point P_e in Fig. 3b is the edge point, and μ_g denotes the geometric center position. In addition, $d(P_e)$ is the distance from the P_e to the geometric center μ_g, and r is the radius of the spherical neighborhood. The symbols in Fig. 3c have similar meanings, and we obtain $RCD(P_e) = 0.6076$ and $RCD(P_i) = 0.1069$. The importance of point P_e is greater than that of point P_i, consistent with human perception of visual importance. Therefore, we use RCD values to represent that the importance of edge points is greater than the interior points. Meanwhile, the proposed RCD can maintain an object's translation, rotation, and scale invariance. In addition, geometric saliency can be computed easily and efficiently, which can be applied to large-scale point cloud processing tasks.

The proposed geometric saliency can be regarded as a kernel function that reflects the geometric properties of a 3D point cloud. Other kernel functions can be introduced into our FL3K method, which indicates that our method provides a generalized framework for 3D keypoint detection. We introduce kernel functions to define a more general representation of geometric saliency, which can be expressed as

$$S_{geo}(p) = \frac{\left\| \frac{\sum_{j=1}^{N(p)} p_j * g\left(\left\| \frac{p - p_j}{r} \right\|^2 \right)}{\sum_{j=1}^{N(p)} g\left(\left\| \frac{p - p_j}{r} \right\|^2 \right)} - p \right\|}{r}, \tag{3}$$

where $g(\cdot)$ denotes the negative of the derivative of the kernel function, and $N(p)$ represents all neighboring points of the point p in a spherical neighborhood of radius r. Here, the neighborhood radius r can be regarded as the bandwidth of the kernel function. The RCD kernel function can be considered as $g(u) = 1$, which is geometrically equivalent to

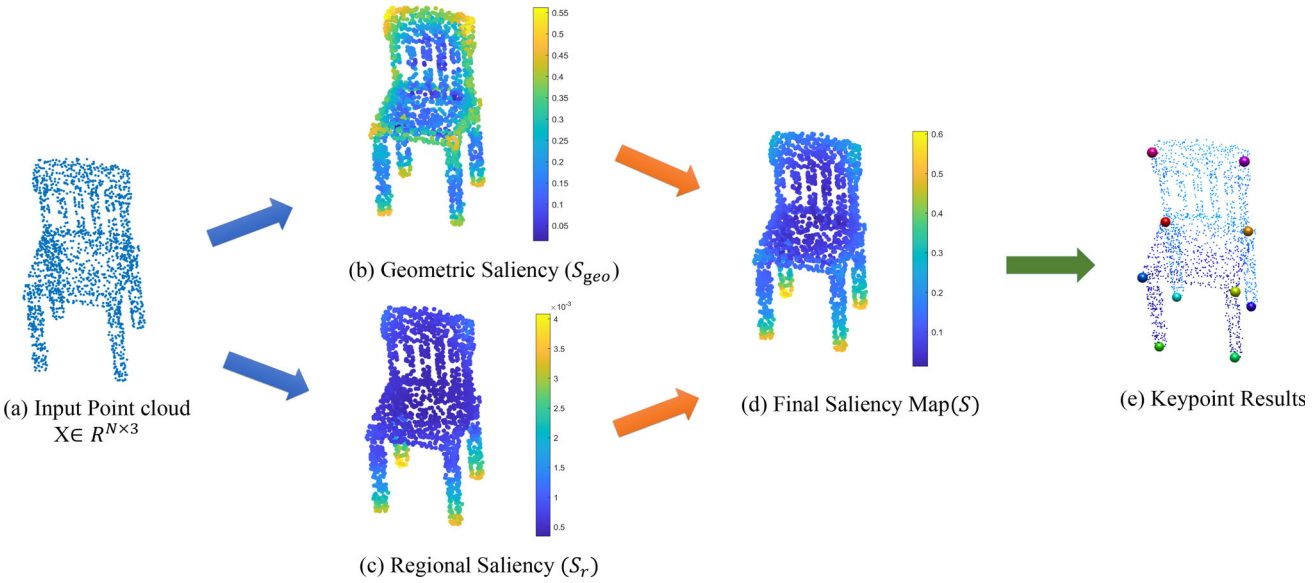


Fig. 2 Outline of the FL3K detector. For an input point cloud X, geometric saliency is calculated to identify small geometric features. Then, regional saliency describes larger regions to capture the distinctness of entire semantic parts. The geometric and regional saliency maps are integrated to generate the final saliency map. Finally, keypoints are generated based on the final saliency map

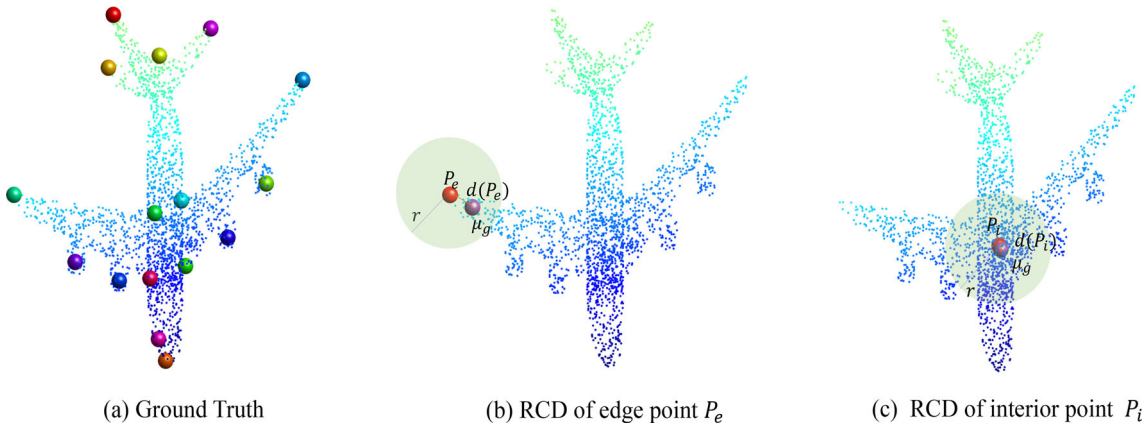


Fig. 3 Comparison of RCDs between an edge point and an interior point. **a** Ground-truth; **b** $RCD(P_e) = 0.6076$; **c** $RCD(P_i) = 0.1069$. Since $RCD(P_e)$ is greater than $RCD(P_i)$, the importance of point P_e is greater than that of point P_i

assuming that the contribution of all points in the neighborhood is the same. We can also use other kernel functions to define the geometric saliency. We present the results of geometric saliency analysis based on different kernel functions in the ablation study in Sect. 4. We chose to use the relative center distance as a kernel function representation because this representation is computationally efficient for 3D keypoint detection.

3.3 Regional Saliency

Inspired by the hierarchical human visual perception mechanism (Grill-Spector & Malach, 2004), we identify globally distinct features in a multi-level manner. The low-level representation is used to detect subtle features while suppressing 3D textures. On the other hand, the regional representation is used to identify the entire unique region. We use geometric saliency as a low-level representation to detect subtle features in point clouds and regional saliency to determine unique regions of point clouds. Since geometric saliency alone is insufficient to characterize points effectively with semantic information, we introduce regional saliency to remedy this deficiency and improve the accuracy of keypoint detection. The regional saliency aims to highlight important parts of the point cloud and obtain regional visual information.

To evaluate the distinctness of entire regions, we use a large neighborhood R. We set $R = 40$ mr in the experiment. We first define the importance of the semantic part for point

p,

$$S_{sp}(p) = mean(S_{geo}(N_R(p))), \tag{4}$$

where $N_R(p)$ represents the set of all neighboring points with a spherical radius R and $mean(\cdot)$ denotes the average value. In this work, $S_{geo}(\cdot)$ computes the geometric saliency of a point. We compute the geometric saliency values of all points within the neighborhood range R to reflect the importance of semantic parts. To calculate geometric saliency, we can use RCD or other kernel functions. We directly use the average geometric saliency values to reflect the semantic importance of the entire region, which is effective for 3D keypoint detection. Figure 2c shows that the four corners of the chair have high semantic importance, which is consistent with what humans perceive. Finally, we exploit the importance of the semantic parts to obtain the regional saliency as

$$S_r(p) = 1 - \exp\left(-\frac{1}{|N_R(p)|} S_{sp}(p)\right). \tag{5}$$

This regional saliency reflects the overall changes in the point cloud by considering the average geometric saliency values of points within the neighborhood range. It avoids the negative influence of noisy points on the overall significance. Here, we only use the relative center distance saliency as the geometric saliency; other geometric saliencies, such as curvature and the normal vector's angle, can be included.

We present an example to show that geometric saliency may obtain wrong keypoints, but they can be removed by introducing regional saliency, as shown in Fig. 4. The keypoints obtained when returning using only geometric saliency may contain erroneous keypoints, as shown in Fig. 4a. The green ellipse in Fig. 4a surrounds the erroneous keypoints. After adding regional saliency to Fig. 4a, the erroneous keypoints are filtered out, as shown in Fig. 4b. Note that the results obtained after removing the incorrect keypoints using regional saliency agree closely with the ground truth. Therefore, the proposed regional saliency can further improve the accuracy of keypoint detection. More discussion on ablations is presented in Sect. 4.8.

3.4 Generation of Stable Keypoints

Geometric and regional saliency can be integrated using a supervised approach or a simple aggregation strategy to extract sable key points. The supervised approach involves labeled data and models such as conditional random fields (Lafferty et al., 2001) and logistic regression (Hosmer et al., 1997). However, it is not practical for 3D keypoint detection as obtaining large-scale labeled data is labor-intensive and time-consuming. Thus, we use an accumulation strategy to achieve geometric and regional saliency fusion.

We present a simple and effective weighted nonlinear suppression aggregation method to combine geometric and regional saliency. The aggregation method effectively suppresses similar peaks in many saliency maps and promotes saliency maps with fewer peak values. Each saliency map S_i is first normalized. We use min-max normalization on the saliency map S_i between [0, 1]. Then, we calculate the maximum saliency M_i and the average m_i of all saliency values except the maximum value. Finally, we multiply the saliency map S_i by $(M_i - m_i)^2$ to obtain the suppressed saliency map S_i'. A nonlinear suppression operation on the geometric and regional saliency maps obtains their suppressed saliency maps, S'_{geo} and S'_r. Finally, we weigh the two saliency maps S'_{geo} and S'_r to obtain the final saliency map S. Therefore, the weighted nonlinear suppression aggregation method is represented as

$$S(p) = w_1 S'_{geo}(p) + (1 - w_1) S'_r(p), \tag{6}$$

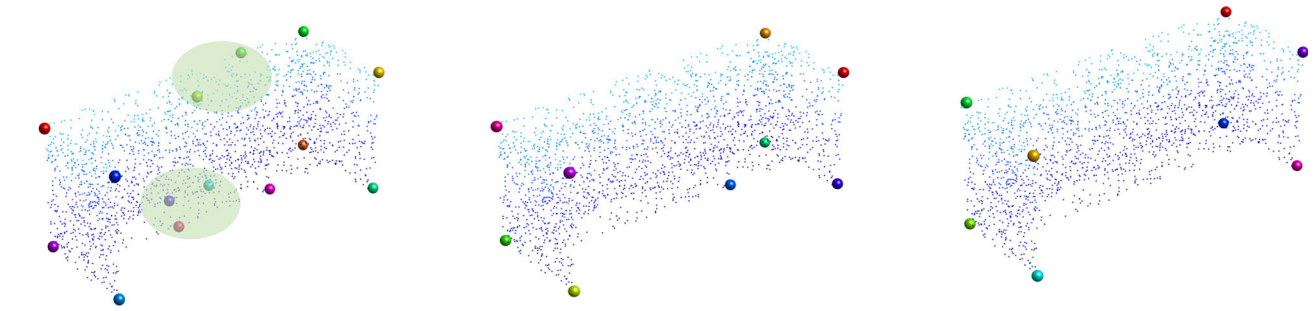
where w_1 is the weight parameter used to balance the importance of geometric and regional saliencies. We set $w_1 = 0.5$ in the experiments, indicating that the types of saliency are equally important in keypoint detection. The saliency map S can be obtained through weighted nonlinear suppression aggregation. Based on this saliency map, we then generate stable keypoints. The average of all the significant values in S is set as a threshold. Furthermore, ln is set as the local neighborhood. If the saliency value of a point is less than the threshold, it is not a keypoint. If the saliency is greater than the threshold and the saliency of all the points in the local neighborhood ln, then it is a keypoint. Algorithm 1 summarizes the process of keypoint generation. This method obtained the keypoint detection results shown in Fig. 2e.

The detected keypoint locations are consistent with those human vision perceives. These results show that our method effectively detects geometrically and semantically consistent keypoints.

4 Experiments

4.1 Experimental Setups

We conduct experiments on four datasets, KeypointNet (You et al., 2020), ShapeNet-Chair (Yi et al., 2017), SMPL (Loper et al., 2015), and Redwood (Choi et al., 2015), to evaluate the performance of FL3K against existing methods. The KeypointNet and Shape-Net-chair datasets contain rigid 3D object models. The SMPL dataset contains nonrigid 3D human models, and Redwood is an RGB-D reconstruction dataset for indoor scenes. There are four parameters in the FL3K method: spherical radius r, large neighborhood R, local neighborhood ln, and weight parameter w_1. In our



(a) Keypoints by the Geometric Saliency (b) Adding Regional Saliency to (a) (c) Ground Truth

Fig. 4 An example showing the importance of regional saliency. The erroneous keypoints in the green ellipse in **a** can be removed by adding regional saliency, as shown in **b**. In addition, the keypoints obtained after adding regional saliency are consistent with those annotated in the ground truth (**c**) (Color figure online)

Algorithm 1 Generation of Keypoints

Input: saliency map S, point cloud X.
Output: a set of keypoints $\widetilde{X}$
1: Initialization: $\widetilde{X} \leftarrow \{\}$, threshold $\eta = mean(S)$.
2: Using KD-tree to calculate the mr of the X and the neighbor N_g in $ln = 10$ mr.
3: **for** each $p \in X$ **do**
4: Extract the saliency value $S(p)$ of point p.
5: **if** $S(p) < \eta$ **then**
6: continue
7: **end if**
8: maximum $\leftarrow$ true
9: Extract the neighbor points $N_g(p)$ for p
10: **for** each $q \in N_g(p)$ **do**
11: **if** $S(p) < S(q)$ **then**
12: maximum $\leftarrow$ false
13: **end if**
14: **end for**
15: **if** maximum is true **then**
16: $\widetilde{X} \leftarrow \widetilde{X} \cup \{p\}$
17: **end if**
18: **end for**

experiments, $r = 15$ mr, $R = 40$ mr, and $ln = 10$ mr, where mr represents the resolution of a point cloud. In addition, we set the weight $w_1 = 0.5$ for performance evaluation. The code for our method is publicly available at https://github.com/zhuanjia113/FL3K.

We use the mean intersection over union (mIoU) performance metric for 3D keypoint detection, defined as the ratio of the number of intersection to union points. The intersection is the set of all points where the geodesic distance between the detected keypoints and the nearest ground truth is less than a given geodesic threshold. The union is the set of detected and ground-truth keypoints. On the KeypointNet and ShapeNet-chair datasets, we compare the predicted keypoints with the ground truth and compute the mIoU values under the threshold. Due to the lack of manually labeled keypoints, we compute mIoU for the SMPL and Redwood datasets by comparing the consistency of keypoints for each pair of 3D objects. A keypoint is referred to as semantically consistent in the first model if the distance between its corresponding point and the nearest keypoint in the second model is below some threshold.

4.2 3D Keypoint Datasets

KeypointNet Dataset The KeypointNet dataset is widely used to evaluate the performance of 3D keypoint detectors. This dataset annotates 8,328 3D models from 16 object categories, with 83,231 manually labeled keypoints. Figure 5a shows a few 3D models from this dataset and the labeled 3D keypoints. Because the objects in this dataset undergo large deformations, detecting 3D keypoints can be challenging.
ShapeNet-Chair Dataset The ShapeNet-Chair (Yi et al., 2017) contains thousands of keypoints labeled by human experts. This dataset includes 1249 3D chair models, and Fig. 5b shows examples from this dataset. As there are large intra-class variations in object appearance, it is difficult to detect all the keypoints.
SMPL Dataset The SMPL dataset is a skinned vertex-based 3D mesh model that captures various pose changes of the human body. Using the farthest point sampling method, we sample 2048 points of the human mesh model for keypoint detection tasks. Figure 5c shows examples from this dataset. These models have large pose variations and complex deformations that other methods cannot detect.
Redwood Dataset The Redwood is an RGB-D reconstruction dataset of indoor scenes. The scene model of this dataset contains a large amount of point cloud data. With the same setup as (Zhong et al., 2022), we use this dataset to evaluate the repeatability of our method. A few samples from this dataset are presented in Fig. 5d. These scene models have complex variations, so the dataset is suitable for evaluating the repeatability of various keypoint detectors.

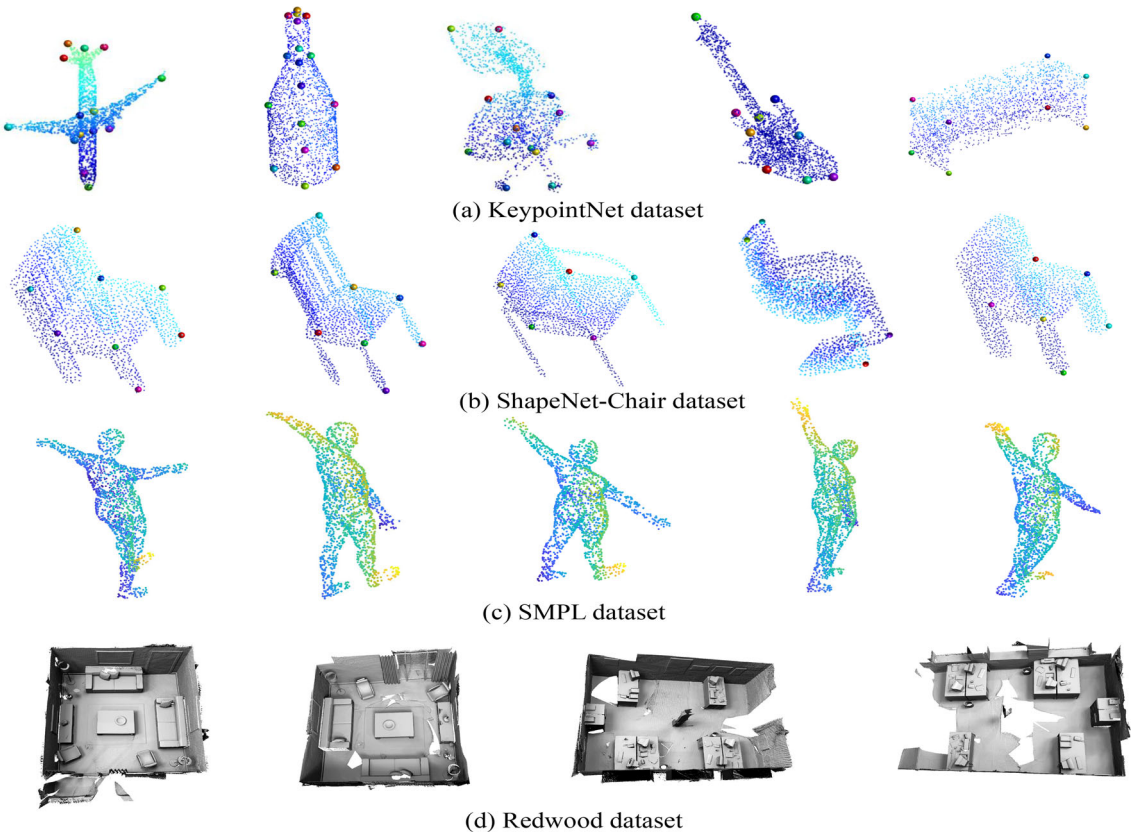


Fig. 5 Sample images of the KeypointNet, ShapeNet-Chair, SMPL, and Redwood datasets. The objects in **a** exhibit a large deformation, the objects in **b** have a large intra-class variation, the objects in **c** have a substantial non-rigid deformation, and the objects in **d** have a complex scene

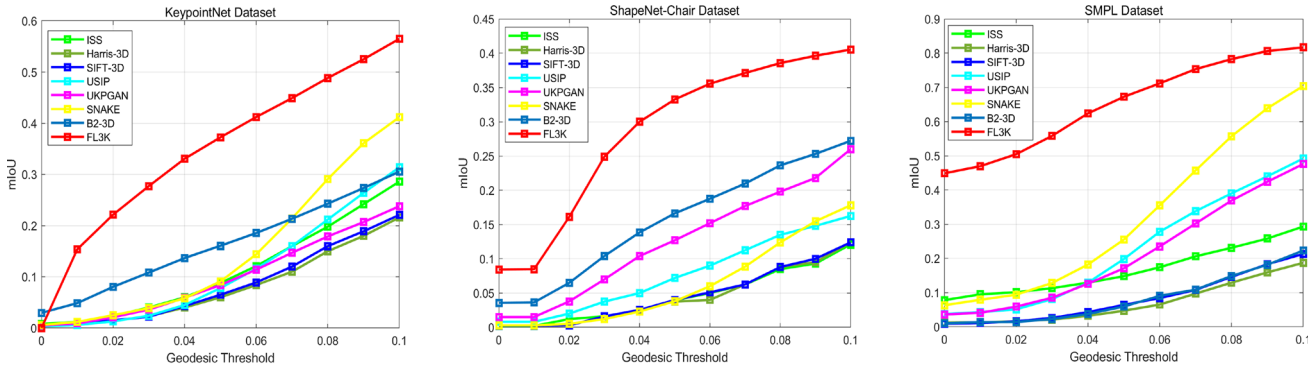


Fig. 6 Experimental results of different methods on the KeypointNet, ShapeNet-Chair, and SMPL datasets

4.3 Keypoint Detection

We evaluate the performance of the FL3K method on the KeypintNet dataset against the Harris-3D (Sipiran & Bustos, 2011), SIFT-3D (Rister et al., 2017), ISS (Zhong, 2009), USIP (Li & Lee, 2019), UKPGAN (You et al., 2022), SNAKE (Zhong et al., 2022), and B2-3D (Wimmer et al., 2024) methods. Harris-3D, SIFT-3D, and ISS are classic keypoint detection methods, while USIP, D3Feat, UKGPAN, SANKE, and B2-3D are deep learning-based methods. In addition, the results of methods other than ours and B2-3D are from the literature. We use the original implementation of the B2-3D method for experiments. A comparison of the evaluations of all methods on the KeypointNet dataset is shown in Fig. 6a. The FL3K method performs favorably against all other evaluated approaches. These results demonstrate the effectiveness of our method in combining geometric saliency with regional saliency.

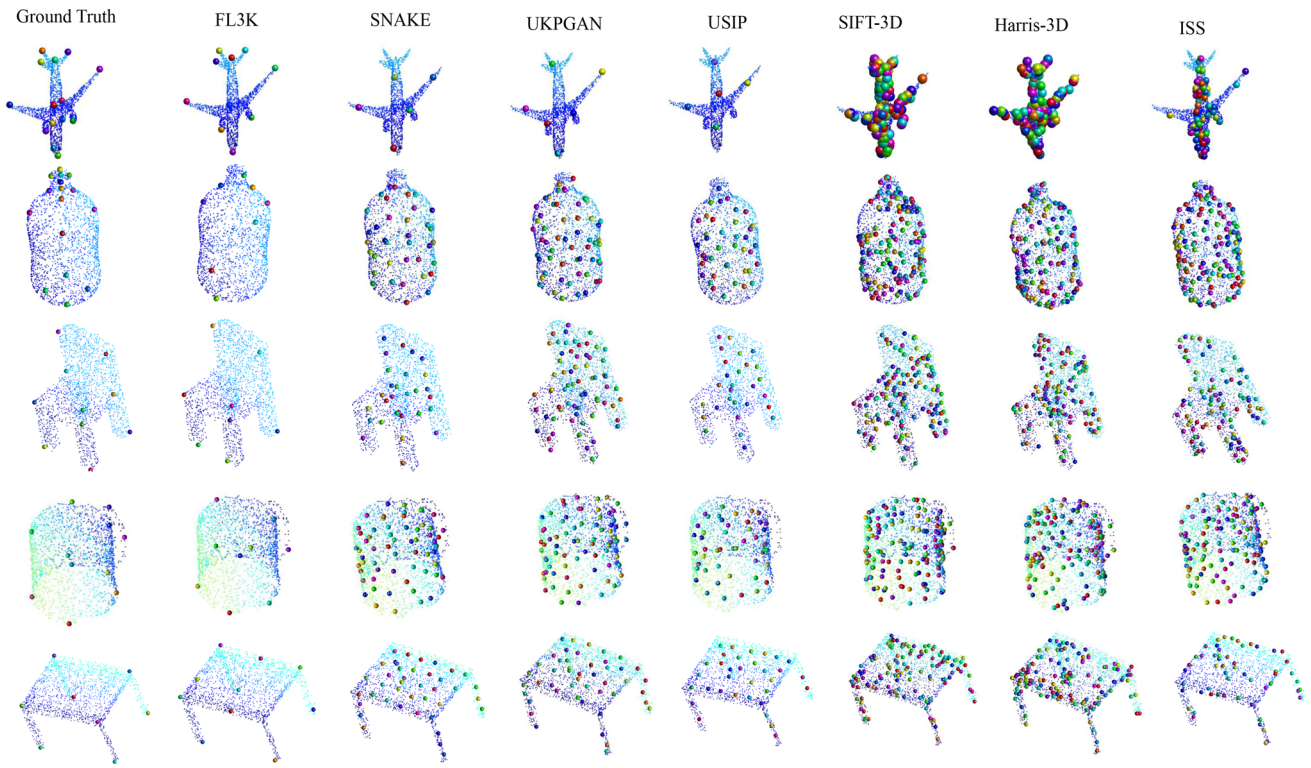


Fig. 7 Visualizations of seven keypoint detection methods on the KeypointNet dataset. The keypoints detected by our FLK detector are closer to the ground truth than other methods

Next, we evaluate our FL3K method on the ShapeNet-Chair dataset. Figure 6b shows that FL3K effectively detects the keypoints of 3D objects with large intra-class appearance changes than other schemes.

Figure 6c shows the evaluation results on the SMPL dataset. FL3K achieves the highest detection results for non-rigid 3D objects.

Figure 7 shows sample detection results of all evaluated methods on the KeypointNet dataset. The keypoints detected by FL3K are corners or points with semantic information consistent with human perception. Other methods either do not detect the keypoints correctly, or the detected keypoints are insignificant and inconsistent with human visual perception. Therefore, our method gives better results for keypoint detection. Figure 8 shows the detection results on the SMPL dataset. The FL3K method can detect keypoints with semantic significance, such as the head, hands, and feet. Other methods detect inconspicuous keypoint positions inconsistent with human visual perception.

4.4 Repeatability

Repeatability is the ability of a detector to detect keypoints at the same position under various interferences, such as viewpoint changes, noise, and missing parts. It is often regarded as the most important metric to measure the stability and robustness of a keypoint detector. We evaluate the stability of the keypoint detection method under the influence of direction changes, noise, and missing parts interference of a point cloud. We use the relative repeatability in the USIP (Li & Lee, 2019) and SNAKE (Zhong et al., 2022) methods as the evaluation metrics. Given two point clouds $\{X, \hat{X}\}$ of a scene captured different viewpoints, a keypoint in the first point cloud X is repeatable if its distance to the nearest keypoint in the other point cloud $\hat{X}$ is below a threshold ϵ. Relative repeatability indicates the number of repeatable keypoints divided by the number of detected ones.

Figure 9 provides an example of the keypoints of repeatability and relative repeatability. The keypoints q_1, q_2, and q_3, connected by green lines in the figure, represent the repeatable points of the point cloud X. These points are transformed through ground-truth rotation and the translation matrix. Their distances to the keypoints in the second point cloud $\hat{X}$ are less than the threshold ϵ. Here, we set the threshold ϵ equal to 0.02. These repeatable points are located at the same positions as the two 3D objects. For example, q_2 in point cloud X is located at the same semantic position of the chair as the corresponding point in the point cloud $\hat{X}$. Relative repeatability is the number of found repeatable points divided by the detected keypoints. Sixteen keypoints are detected in point cloud X, with three repeatability points. The relative repeatability of point cloud X is 18.75%.

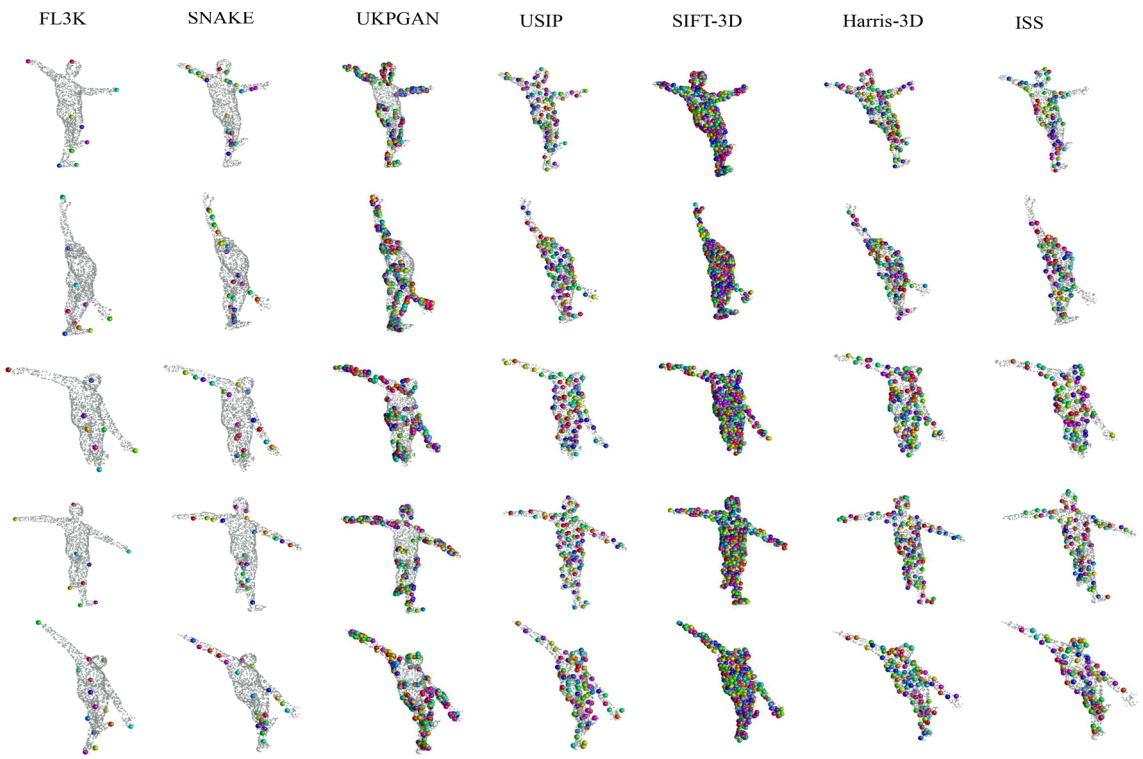
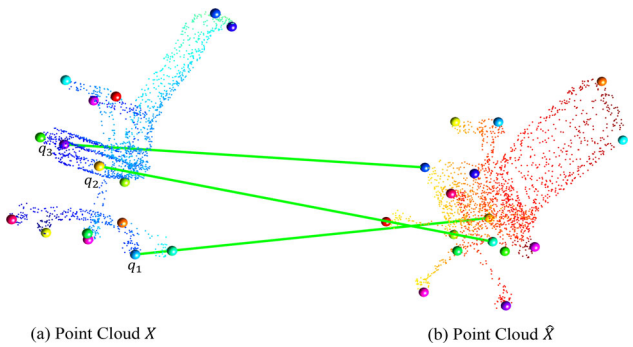


Fig. 8 Visualization results of different methods on the SMPL dataset. The keypoints detected by our FL3K detector are sparse, semantically important, and highly repeatable compared to other methods



(a) Point Cloud X (b) Point Cloud $\hat{X}$

Fig. 9 An example of repeatability keypoints. The three points q_1, q_2, and q_3 connected by green lines represent the repeatable keypoints of the point cloud X because the distance from these three points to the corresponding points in the point cloud $\hat{X}$ after the ground truth transformation is less than the threshold ϵ (Color figure online)

We conduct repeatability experiments using the KeypointNet and Reedwood datasets. Using the same settings as for the SNAKE (Zhong et al., 2022) method, we generate 64 keypoints for each point cloud from the Redwood dataset and obtain the detection results with different thresholds, downsampling rates, and Gaussian noise. Figure 10 shows the repeatability of all the evaluated methods. Our method performs well when the keypoints are rotated because the RCD representation is invariant to the similarity changes of a point cloud. Although our method's relative repeatability decreases significantly with larger downsampling, it still outperforms the other schemes at the maximum downsampling rate. As shown in Fig. 10c, our method performs well when Gaussian noise is added to the point clouds. The FL3K method achieves more than 70% repeatability for $\sigma = 0.1$, significantly better than the other approaches.

Next, we evaluate the repeatability of our method on the KeypointNet dataset. Using the same setup as for the SNAKE (Zhong et al., 2022) method, we select the 32 most salient key points for each point cloud to evaluate the method's robustness. We present the detection results of our method under arbitrary rotation variations, down-sampling rates, and Gaussian noise. Table 1 shows that FL3K performs better than other schemes for arbitrary rotation variation. ISS also performs well under various rotations. Table 2 shows that our method performs best under different downsampling rates and noise.

4.5 Keypoints for Real-World Datasets

Our proposed FL3K keypoint detector can be applied to many tasks in computer vision, such as 3D reconstruction, target localization and recognition, and point cloud-based SLAM. To illustrate its potential application value and validate our method's efficacy and robustness, we conduct point cloud registration experiments on the 3DMatch (Zeng et al.,

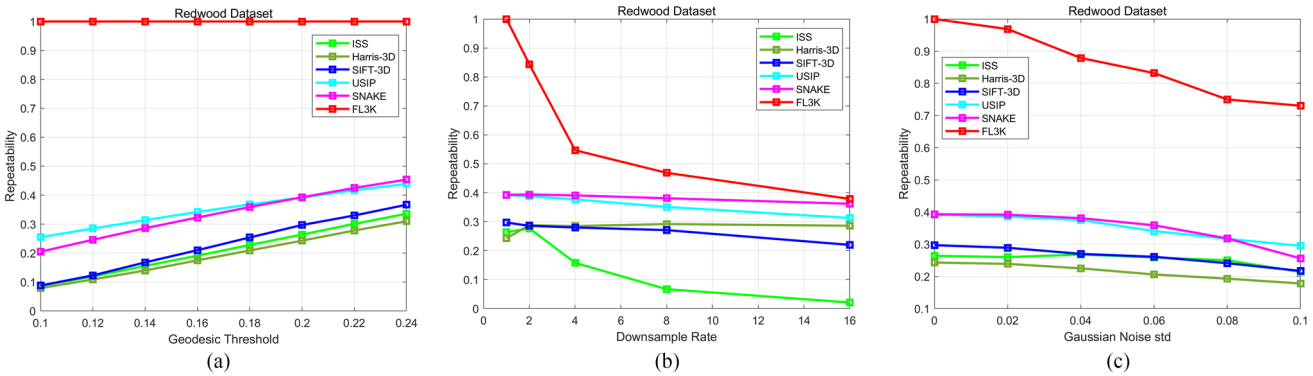


Fig. 10 Relative repeatability for two-view point clouds with different distance threshold (**a**), down-sampling rate (**b**), Gaussian noise $N(0, \sigma)$ (**c**) on Redwood

Table 1 Relative repeatability with different distance thresholds ϵ on the KeypointNet dataset

Method	$\epsilon = 0.03$	$\epsilon = 0.05$	$\epsilon = 0.07$	$\epsilon = 0.09$	$\epsilon = 0.1$
ISS (Zhong, 2009)	0.9846	0.9935	0.9977	0.9989	0.9986
UKPGAN (You et al., 2022)	0.199	0.454	0.661	0.810	0.864
SNAKE (Zhong et al., 2022)	0.643	0.806	0.892	0.936	0.948
FL3K	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**

The bolded numbers indicate the best performance

Table 2 Relative repeatability when input point clouds are disturbed ($\epsilon = 0.03$)

Method	Original	$\gamma = 4$	$\gamma = 8$	$\sigma = 0.02$	$\sigma = 0.03$
ISS (Zhong, 2009)	**1.0**	0.1282	0.0602	0.3246	0.1838
UKPGAN (You et al., 2022)	0.199	0.570	0.427	0.608	0.558
SNAKE (Zhong et al., 2022)	0.643	0.594	0.525	0.626	0.536
FL3K	**1.0**	**0.7150**	**0.5538**	**0.8425**	**0.7213**

Here, γ is the downsample rate. The bolded numbers indicate the best performance

2017) and ETH (Pomerleau et al., 2012) datasets. 3DMatch is a dataset of indoor scenes used as a benchmark for point cloud registration experiments. This dataset contains eight test scenes with partially overlapping point cloud data. It also provides a ground-truth transformation matrix that can be used to evaluate the registration performance of various keypoint detectors. The ETH dataset is a registration benchmark for outdoor scenes, and the test set from this dataset contains four scenes with overlapping parts. Our method does not require training, and we perform registration experiments directly on 3DMatch and ETH test scenes.

We follow a previous method (You et al., 2022) to set a voxel grid filter of 0.03 m and 0.02 m for downsampling the point cloud data on the 3DMatch and ETH datasets. Meanwhile, we also use three standard metrics: feature matching recall (FMR), registration recall (RR), and inlier ratio (IR). FMR represents the percentage of successful alignments whose IR is above a threshold (i.e., $\tau_1 = 5\%$) that measures the matching quality of pairwise registration. RR is the percentage of successful alignments whose transformation error is below a threshold (i.e., RMSE < 0.2 m) that reflects the final performance in practice. For a pair of point clouds, the number of matching points is M, and a matching point is considered an inlier if the distance between its corresponding points is smaller than $\tau_2 = 0.1$ m under ground-truth transformation. If the total number of inliers is Q, then its IR is $\frac{Q}{M}$. We use the registration performance for each point cloud when the number of returned keypoints is 100. The fewer the keypoints there are, the more it reflects the robustness and efficiency of the keypoint detector, thus enabling faster processing of large-scale point clouds. Point cloud registration consists of two steps: keypoint detection and descriptor extraction. Here, we directly use the off-the-shelf descriptor D3Feat (Bai et al., 2020) as the feature representation of the point cloud; the same features are used for all other keypoint detectors. In particular, we compare FL3K with other methods (Rister et al., 2017; Teng et al., 2023; You et al., 2022; Zhong, 2009; Zhong et al., 2022). Tables 3 and 4 compare our method's results to other methods on 3DMatch and ETH, respectively.

Table 3 shows that FL3K achieves the best registration performance on the 3DMatch dataset, surpassing traditional

Table 3 Registration result on 3DMatch

Detector	Descriptor	FMR (%)	RR (%)	IR (%)
Random	D3Feat	81.2	38.8	17.3
ISS (Zhong, 2009)	D3Feat	81.0	37.2	17.4
SIFT-3D (Rister et al., 2017)	D3Feat	81.3	38.6	17.4
UKPGAN (You et al., 2022)	D3Feat	85.9	47.4	27.7
SNAKE (Zhong et al., 2022)	D3Feat	89.5	50.9	30.0
CED-3D (Teng et al., 2023)	D3Feat	99.44	47.28	16.14
FL3K	D3Feat	**100**	**58.27**	**39.45**

We combine the D3Feat feature and different keypoint detectors to perform point cloud registration. The bold numbers represent the best performance and the underlined numbers indicate second-ranked performance

Table 4 Registration result on ETH

Detector	Descriptor	FMR (%)	RR (%)	IR (%)
Random	D3Feat	2.1	1.5	6.3
ISS (Zhong, 2009)	D3Feat	6.2	1.7	7.5
SIFT-3D (Rister et al., 2017)	D3Feat	5.5	1.1	6.7
UKPGAN (You et al., 2022)	D3Feat	21.5	3.9	9.2
CED-3D (Teng et al., 2023)	D3Feat	55.0	1.33	6.23
FL3K	D3Feat	**81.26**	**4.39**	**10.03**

We combine the D3Feat feature and different keypoint detectors to perform point cloud registration. The bold numbers represent the best performance and the underlined numbers indicate second-ranked performance

(Rister et al., 2017; Zhong, 2009) and deep learning keypoint detectors (You et al., 2022; Zhong et al., 2022). It achieves 100% FMR, demonstrating its high stability on the keypoint detection task with an inlier ratio greater than the threshold τ_1 in all pairwise point cloud registrations. Both traditional (Rister et al., 2017; Zhong, 2009) and deep learning methods (You et al., 2022; Zhong et al., 2022) have inlier ratios below τ_1, indicating these methods' instability in some pairwise point cloud registrations. Our method outperforms the recent CED-3D method (Teng et al., 2023) by 0.6%, 11%, and 23% in FMR, RR, and IR metrics, respectively. These results show that FL3K is effective in the registration of indoor scenes. Our method holds immense potential for point cloud registration tasks.

Table 4 shows that our FL3K method demonstrates the best registration performance on the outdoor ETH dataset, outperforming the recent CED-3D method (Teng et al., 2023) by 26%, 3%, and 3.8% in FMR, RR, and IR metrics. Note that all the methods perform poorly in registration on this dataset due to the complex variability of the point cloud data. However, our method still outperforms traditional and deep learning methods. These results show that FL3K is also effective for registration tasks in outdoor scenes. In addition, we present examples of keypoint matching from the 3DMatch and ETH datasets in Fig. 11. The red dots in the figure indicate the detected keypoints, and the green lines indicate the matching results between the found keypoints. IR indicates the inlier ratio of matches between each pair of point clouds. FL3K can still find the matching results of keypoints correctly. Notice that the keypoints detected by our method are also more consistent with our human visual perception. In addition, The IR value of our method in 3DMatch is larger than the IR value in ETH, which may be due to the complexity of objects in the outdoor scene changes more than in the indoor scene. Our method can detect the corresponding matching keypoints in complex indoor and outdoor scenes, showing its effectiveness for point cloud registration tasks.

4.6 Keypoint Detection Under Partial Occlusion

We conduct experiments on partially occluded point clouds. We randomly generate the location to be occluded and remove the points within the local neighborhood of that location to generate the occluded point cloud data. For experiments, we occlude 10%, 20%, 30%, 40%, and 50% of the keypoints on the KeypointNet dataset. Figure 12 shows an example of point cloud data with different occlusion rates.

We evaluate the keypoint detection performance of our method under different occlusion rates. Table 5 presents the comparison results of our method under different occlusion rates on the KeypointNet dataset. If the occluded part contains a labeled ground truth point, it is not considered for calculating the mIOU value. Our method performs robustly against partial occlusion, especially at low distance thresholds. For example, the detection accuracy of our method decreases by only 4% for a distance threshold $\epsilon = 0.02$ at 50% occlusion. The performance of our method at 50%

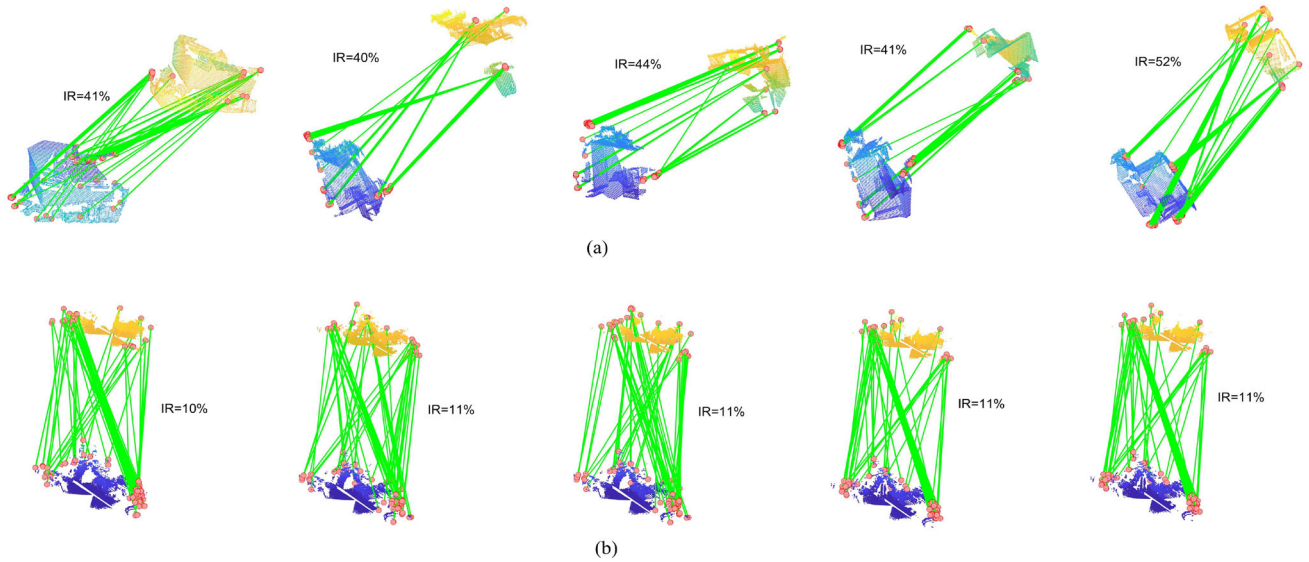


Fig. 11 Some keypoint detection and matching examples of the FL3K detector are from the 3DMatch (**a**) and ETH (**b**) datasets. The red dots indicate the detected keypoints, the green lines indicate the matching results between the found keypoints, and IR indicates the inlier ratio of matches between two point clouds. Our FL3K detector can still correctly find the matching results of keypoints (Color figure online)

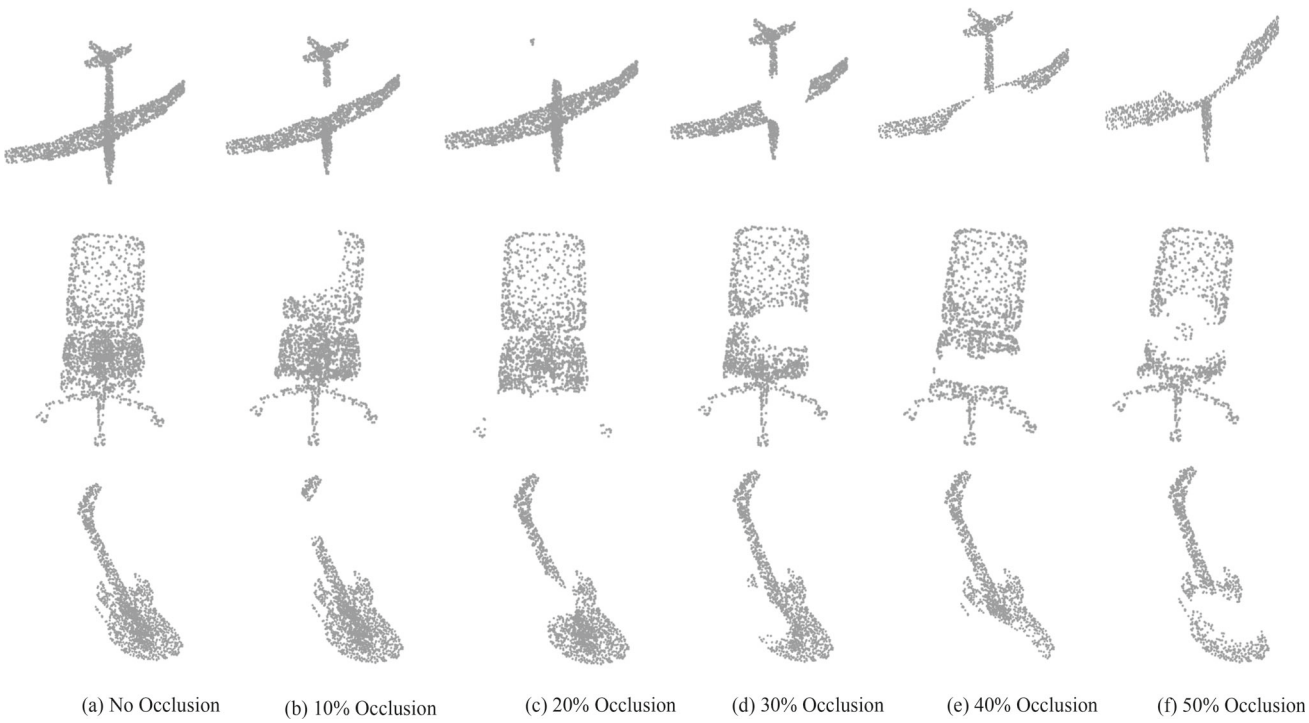
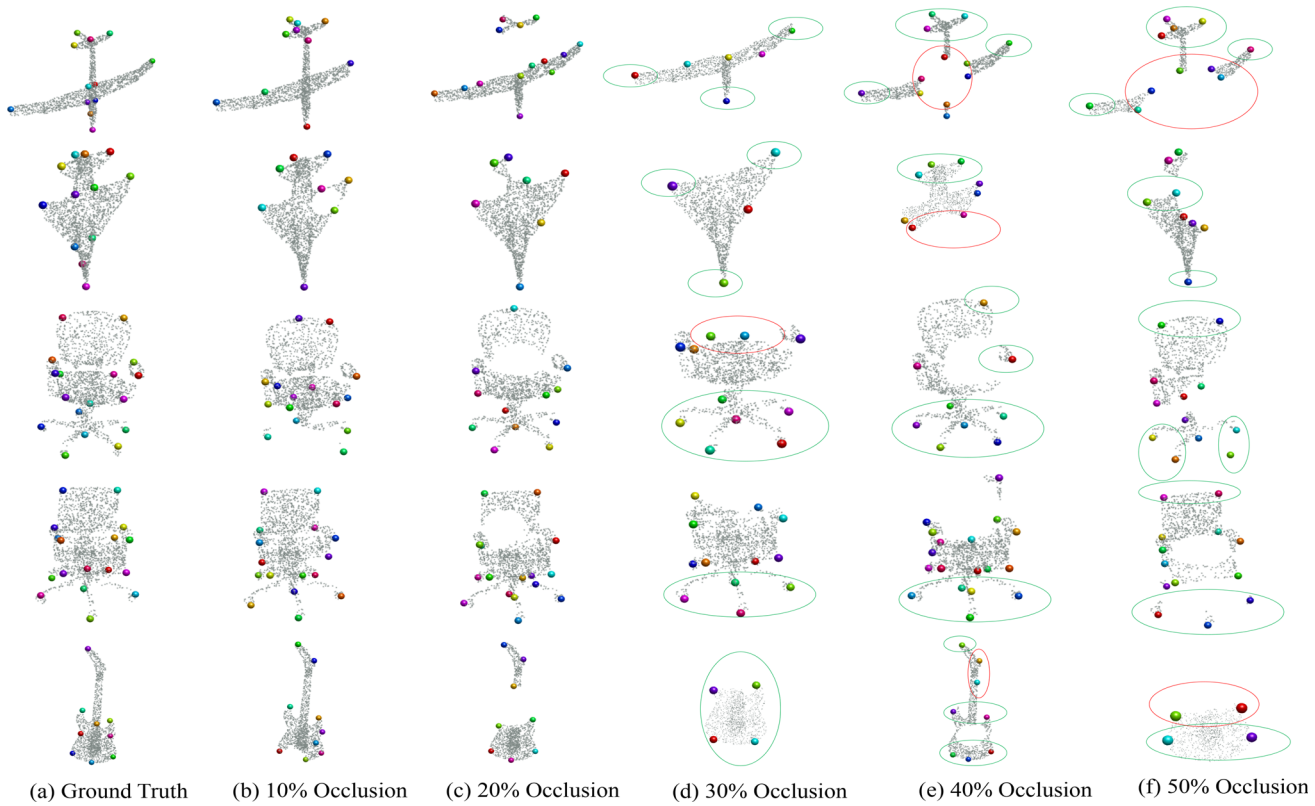


Fig. 12 Point cloud data under different occlusion rates based on the KeyPointNet dataset. From left to right, the occluded parts in the 3D model gradually increase

Table 5 mIoU values of our method with different occlusion rates on the KeypointNet dataset

occlusion rates	$\epsilon = 0.02$	$\epsilon = 0.04$	$\epsilon = 0.06$	$\epsilon = 0.08$	$\epsilon = 0.1$
No occlusion	**0.2214**	**0.3307**	**0.4122**	**0.4885**	**0.5649**
10%	0.2073	0.3051	0.3799	0.4526	0.5254
20%	0.2008	0.2949	0.3665	0.4351	0.5028
30%	0.1922	0.2789	0.3487	0.4176	0.4828
40%	0.1856	0.2657	0.3319	0.3959	0.4590
50%	0.1792	0.2545	0.3178	0.3797	0.4404

The bolded numbers indicate the best performance



(a) Ground Truth (b) 10% Occlusion (c) 20% Occlusion (d) 30% Occlusion (e) 40% Occlusion (f) 50% Occlusion

Fig. 13 Keypoint detection results of the FL3K detector on the KeypointNet dataset under different occlusion rates. The green and red ellipses indicate correctly and incorrectly detected keypoints, respectively, and our FL3K detector can still detect the correct keypoints under high occlusion rates (Color figure online)

occlusion rate is better than other keypoint detection schemes on the KeypointNet dataset.

Figure 13 shows keypoint detection results under different occlusion rates on the keypointNet dataset. The green ellipses in the figure indicate correctly detected keypoints, and the red ellipses represent incorrectly detected keypoints. The FL3K method can still detect the correct keypoints under high occlusion rates. For example, the leg part of the chair accurately detects the correct keypoints under different occlusion rates. The incorrect keypoints in the red ellipse are inconsistent with the labeled ground truth. We note humans likely consider these points belonging to the keypoints when given only these localized segments. For example, the keypoints labeled by the red ellipses of the last two objects in the first row of airplanes and the last row of guitars. Overall, the keypoints detected by FL3K are more consistent with human visual perception.

4.7 Sensitivity Analysis

We analyze the FL3K parameters in this section. Our method has four parameters: spherical radius r, larger neighborhood R, local neighborhood ln, and weight w_1. We explore the impact of these parameters on the performance of keypoint detection. Further, we use the KeypointNet dataset for parameter analysis and mIoU as the performance metric. The other

parameters remain unchanged when we analyze a particular parameter.

First, we analyze the effect of spherical radius r on our method's performance. We set r to 5, 10, 15, and 20 mr, where mr is the size of the point cloud resolution. Table 6 reports the results from our method under different spherical radii r and geodesic distance thresholds ϵ. Our method performs increasingly better as the value of r increases. The performance change is small when the r value enters a specific range. Our method achieves the best detection at $r = 15$ mr when ϵ is greater than or equal to 0.08 and the best at $r = 20$ mr when ϵ is less than 0.08. However, a larger r increases the time needed for keypoint detection. Therefore, we set $r = 15$ mr for the best balance between performance and efficiency.

Next, we analyze the effect of a larger neighborhood R on FL3K's performance. For larger R, we chose 35 mr, 40 mr, and 45 mr. Table 7 summarizes the results of our method for different R values on KeypointNet. The performance of our method improves as R increases. Moreover, its best performance is at $R = 40$ mr when ϵ exceeds 0.04. When the ϵ value is above 0.04, our method achieves the best detection at $R = 45$ mr. However, the performance difference between our method at $R = 40$ mr and $R = 45$ mr is small at larger thresholds. Thus, we set $R = 40$ mr in the experiment.

Third, we analyze the effect of local neighborhoods on keypoint detection in our method. For local neighborhood ln, we take three values: 5, 10, and 15 mr. Table 8 gives the experimental results of our method for different local neighborhoods ln. The performance of FL3K increases then decreases as ln increases when the distance threshold ϵ exceeds 0.02. Our method is superior when $ln = 10$ mr for a threshold greater than 0.02; it performs best at $ln = 15$ mr for ϵ=0.02. The difference between the performance of our method at $ln = 10$ mr and $ln = 15$ mr is small, so we determine that setting $ln = 10$ mr is most suitable.

Finally, we analyze the effect of the weight w_1 on the experimental results of our method. For the parameter w_1, we take 11 values with intervals of 0.1 between [0,1]. Our method uses only regional saliency to localize the keypoints when $w_1 = 0$, while it uses only geometric saliency to detect keypoints when $w_1 = 1.0$. Table 9 gives the FL3K results for different w_1 values on KeypointNet. The performance of our method increases and then decreases as ϵ increases and achieves its best performance for $w_1 = 0.3$ when ϵ = 0.02. The best performance is obtained $w_1 = 0.4$ when ϵ =0.04 and ϵ =0.06. In addition, our method achieves the best performance for $w_1 = 0.5$ when ϵ =0.08 and ϵ =0.1. Note that the difference between the mIoU value of our method and the best mIoU overall is small for $w_1 = 0.5$ when ϵ is less than or equal to 0.06. Thus, we set w_1 = 0.5 for the best performance balance for different ϵ. These results show that geometric and regional saliency are crucial for keypoint detection. They also show that our method is effective for keypoint detection by combining geometric features with semantic information.

Table 6 mIoU values of our method for different spherical radius r on the KeypointNet dataset

r	$\epsilon = 0.02$	$\epsilon = 0.04$	$\epsilon = 0.06$	$\epsilon = 0.08$	$\epsilon = 0.1$
5 mr	0.1496	0.2374	0.3212	0.4061	0.4844
10 mr	0.2043	0.3049	0.3907	0.4743	0.5560
15 mr	0.2214	0.3307	0.4122	**0.4885**	**0.5649**
20 mr	**0.2294**	**0.3396**	**0.4160**	0.4854	0.5597

The bolded numbers indicate the best performance

Table 7 mIoU values of our method for different values of R on the KeypointNet dataset

R	$\epsilon = 0.02$	$\epsilon = 0.04$	$\epsilon = 0.06$	$\epsilon = 0.08$	$\epsilon = 0.1$
35 mr	0.2206	0.3238	0.4019	0.4765	0.5519
40 mr	**0.2214**	**0.3307**	0.4122	0.4885	0.5649
45 mr	0.2193	0.3299	**0.4135**	**0.4903**	**0.5655**

The bolded numbers indicate the best performance

Table 8 mIoU values of our method with different local neighborhood ln on the KeypointNet dataset

ln	$\epsilon = 0.02$	$\epsilon = 0.04$	$\epsilon = 0.06$	$\epsilon = 0.08$	$\epsilon = 0.1$
5 mr	0.1721	0.2569	0.3238	0.3860	0.4412
10 mr	0.2214	**0.3307**	**0.4122**	**0.4885**	**0.5649**
15 mr	**0.2227**	0.3224	0.3952	0.4619	0.5290

The bolded numbers indicate the best performance

Table 9 mIoU values of our method with different weight w_1 on the KeypointNet dataset

w_1	$\epsilon = 0.02$	$\epsilon = 0.04$	$\epsilon = 0.06$	$\epsilon = 0.08$	$\epsilon = 0.1$
0	0.2104	0.2924	0.3466	0.3961	0.4418
0.1	0.2233	0.3166	0.3775	0.4327	0.4821
0.2	<u>0.2285</u>	0.3270	0.3921	0.4500	0.5041
0.3	**0.2292**	<u>0.3347</u>	0.4067	0.4693	0.5305
0.4	0.2276	**0.3370**	**0.4145**	0.4837	0.5524
0.5	0.2214	0.3307	<u>0.4122</u>	**0.4885**	**0.5649**
0.6	0.2136	0.3177	0.4029	<u>0.4856</u>	<u>0.5648</u>
0.7	0.2040	0.3031	0.3897	0.4756	0.5574
0.8	0.1941	0.2866	0.3738	0.4598	0.5437
0.9	0.1848	0.2729	0.3589	0.4462	0.5323
1.0	0.1723	0.2580	0.3443	0.4329	0.5193

The bolded numbers indicate the best accuracy, and the underlined numbers indicate the second-best accuracy

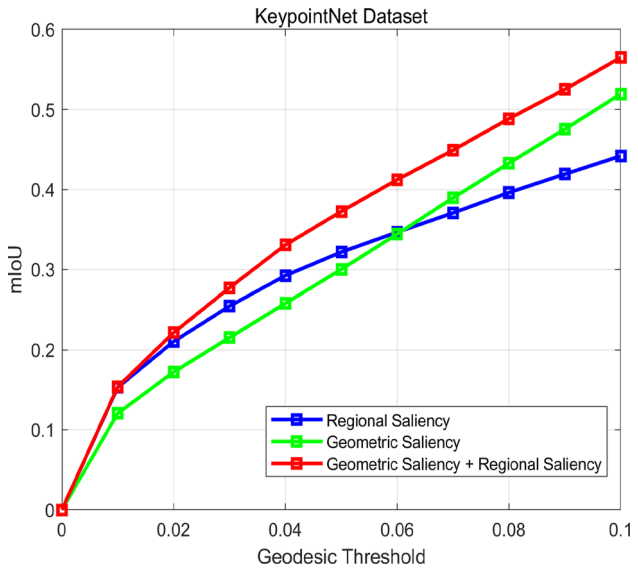


Fig. 14 Comparison of different saliencies on KeypointNet

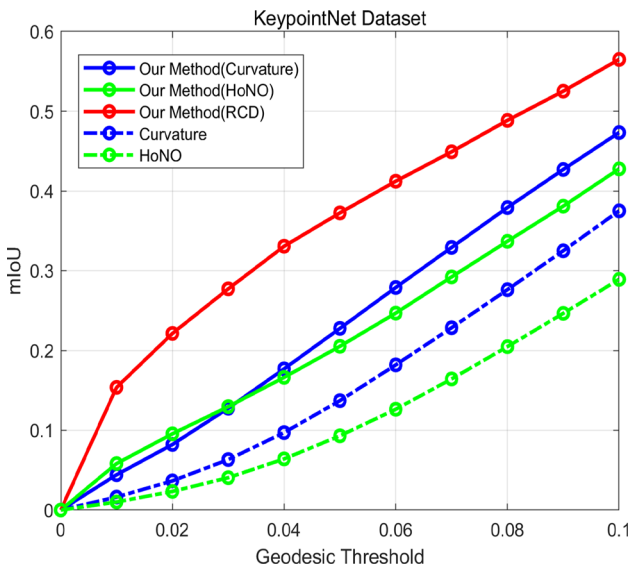


Fig. 15 Comparison of different geometric features on KeypointNet

4.8 Ablation Studies

In this section, we report an ablation study conducted to analyze the impact of different parts of our method on the results. We use the KeypointNet dataset for the analysis and mIOU as the performance metric. Our FL3K method uses geometric and regional saliency for keypoint detection, so we first analyze the performance using only geometric or regional saliency on the keypoint detection task. Figure 14 compares our method with different saliencies. The results show that geometric saliency helps obtain a greater mIoU than regional saliency for geodetic distance thresholds greater than 0.06. The mIoU value is less than saliency for geodetic distance thresholds less than 0.06. Thus, geometric and semantic information is important for 3D keypoint detection. Additionally, FL3K further improves the performance of 3D keypoint detection by combining geometric and regional saliency. FL3K's combination of geometric structure information with the semantic information of a point cloud makes it highly effective in performing keypoint detection tasks.

Due to the use of RCD geometric features in our method, we explore the impact of using different geometric features on the experimental results. We choose the curvature (Taylor, 2023) and histogram of normal orientations(HoNO) (Prakhya et al., 2016) as the geometric feature to compare with the proposed RCD geometric feature. Figure 15 shows the performance of our RCD representation against other geometric features. Our simple RCD geometric features perform better than the curvature and HoNO geometric features on the keypoint detection task. At the same time, the performance per curvature and HoNO geometric features can be further improved by incorporating them with regional saliency. These results demonstrate that the proposed regional saliency is also highly effective for keypoint detection.

Since the proposed RCD representation can be regarded as a kernel function, we analyze different kernel functions in geometric saliency to explore the effect of different kernel functions on the performance of our method. FL3K is a generalized framework for 3D keypoint detection. First, we analyze the performance of geometric saliency based on different kernel functions. We chose to test the Gaussian, Triweight, Cosine, and Epanechnikov kernel functions. Table 10 lists the experimental results of geometric saliency on the KeypointNet dataset for different kernel functions.

The results show the geometric saliency based on the Triweight kernel function helps achieve the best mIoU value when $\epsilon = 0.02$. The geometric saliency based on the Gaussian kernel function achieves the best mIoU value when $\epsilon = 0.04$ and 0.06. The geometric saliency based on the Cosine kernel function achieves the best mIoU value when $\epsilon = 0.08$ and 0.1. Note that the performance difference of geometric saliency based on Gaussian, Cosine, and Epanechnikov functions is small. In addition, the performance difference of geometric saliency based on the RCD and the Triweight is also small. The proposed RCD representation can be regarded as a kernel function with the same performance as the Triweight kernel for keypoint detection tasks. Overall, the performance differences of these geometric saliencies based on different kernel functions are relatively stable, especially when the distance threshold ϵ is less than or equal to 0.08.

Next, we analyze the performance of keypoint detection based on different kernel functions after introducing saliency. The other parameter settings in the experiment remain unchanged, except for differences in geometric and

Table 10 mIoU values of geometric saliency with different kernel functions on the KeypointNet dataset

Kernel	$\epsilon = 0.02$	$\epsilon = 0.04$	$\epsilon = 0.06$	$\epsilon = 0.08$	$\epsilon = 0.1$
RCD	0.1723	0.2580	0.3443	0.4329	0.5193
Gaussian	0.1692	**0.2703**	**0.3589**	0.4498	0.5346
Triweight	**0.1757**	0.2641	0.3464	0.4339	0.5155
Cosine	0.1688	0.2702	0.3588	**0.4499**	**0.5349**
Epanechnikov	0.1650	0.2687	0.3584	0.4493	0.5344

The bolded numbers indicate the best accuracy, and underlined numbers indicate the second best accuracy

Table 11 mIoU values of our FL3K method with different kernel functions on the KeypointNet dataset

Kernel	$\epsilon = 0.02$	$\epsilon = 0.04$	$\epsilon = 0.06$	$\epsilon = 0.08$	$\epsilon = 0.1$
RCD	0.2214	0.3307	0.4122	0.4885	0.5649
Gaussian	0.2260	0.3382	0.4168	0.4886	0.5608
Triweight	**0.2270**	0.3370	**0.4182**	**0.4920**	**0.5661**
Cosine	0.2259	**0.3384**	0.4169	0.4887	0.5609
Epanechnikov	0.2248	0.3367	0.4147	0.4868	0.5591

The bolded numbers indicate the best accuracy and underlined numbers indicate the second best accuracy

regional saliency. Table 11 presents the experimental results for our FL3K method with different kernel functions on the KeypointNet dataset. FL3K performs best in tasks using the Triweight kernel function except for $\epsilon = 0.04$. Our FL3K performs best using the Cosine kernel function when $\epsilon = 0.04$. Note that the performance of FL3K remains stable for different kernel functions, indicating that our method is less affected by the kernel function than others. In addition, we calculate the total time used with different kernel functions, as shown in Fig. 16. The time consumptions are obtained under the same conditions and represent the total time to complete the keypoint detection task for all 3D models in KeypointNet.

The proposed RCD-based geometric saliency is the most efficient in terms of runtime. It demonstrates that the RCD geometric saliency has a high detection efficiency. The geometric saliencies based on the other four kernel functions have similar runtime complexity. The Epanechnikov kernel is approximately 1.3 times the time required by the RCD geometric saliency scheme. We achieve the best balance between efficiency and performance using RCD to represent geometric saliency.

We propose a simple and effective suppression aggregation method to combine geometric and regional saliency. This method facilitates generating saliency maps with few peaks while suppressing others. We use the weighted geometric and regional saliency to generate the saliency map:

$$S(p) = w_1 S_{geo}(p) + (1 - w_1) S_r(p), \tag{7}$$

where w_1 is the weight to balance the importance of geometric and regional saliency, and $S_{geo}(p)$ as well as $S_r(p)$ denote the geometric and regional saliency of a point p, respectively. Next, we compare the performance of the weighted

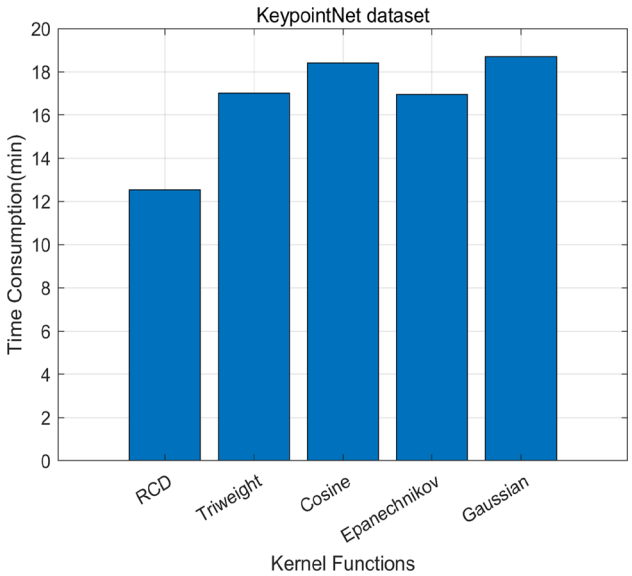


Fig. 16 Time consumed by different kernel functions on KeypointNet

average aggregation method with the nonlinear suppression aggregation method for keypoint detection. Figure 17 shows the results on the KeypointNet dataset. The nonlinear suppression aggregation method helps obtain better detection accuracy than the weighted average aggregation method. If the weighted average accumulation method is used, numerous regions will be marked as salient points, affecting the accuracy of keypoint detection. Figure 18 shows some visualization examples. Figure 18a represents the annotated ground truth, Fig. 18b illustrates the saliency map obtained by weighted average accumulation, and Fig. 18c represents the saliency map obtained by the nonlinear suppression method. Figure 18d represents the keypoint detection results obtained

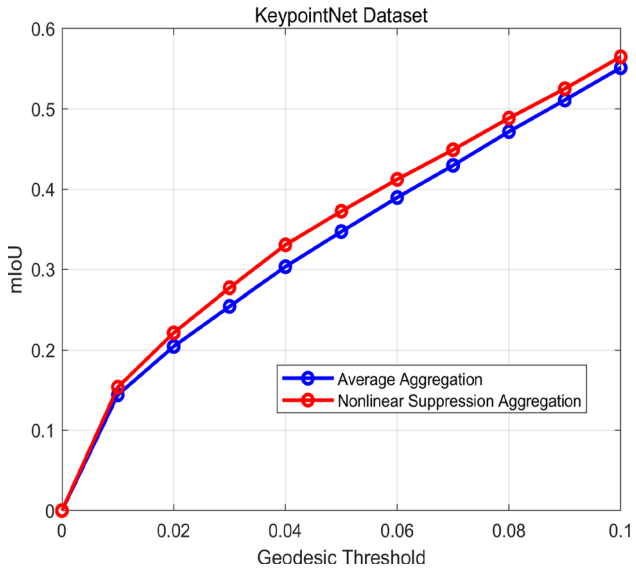


Fig. 17 Comparison of different aggregation methods on KeypointNet

using the weighted average aggregation method, and Fig. 18e represents the keypoint detection results obtained using the nonlinear suppression aggregation method. The nonlinear aggregation method can effectively suppress some erroneous keypoint detection results. For example, the keypoints marked with red ellipses in Fig. 18d are incorrectly detected, while these erroneous keypoints in Fig. 18e have been suppressed. Thus, we use a nonlinear suppression aggregation method to combine geometric and regional saliency to detect keypoints.

4.9 Computational Complexity

We analyze the computational complexity of our FL3K method. Its complexity reflects geometric and saliency computation and keypoint generation. The complexity of geometric saliency is used to construct the RCD feature. The computational complexity of the RCD feature is $O(NK_1)$, where N is the number of points in the point cloud, and K_1 represents the number of neighborhood points with a spherical radius r. Meanwhile, regional saliency has a computational complexity of $O(NK_2)$, where K_2 is the number of points in a large neighborhood R. Therefore, the total time complexity to compute the geometric and regional saliency is $O(NK_1) + O(NK_2) = O(NK_2)(K_1 < K_2)$. When generating keypoints through saliency map S, the computational complexity required is $O(NK_3)$, where K_3 is the number of points in the local neighborhood ln. Therefore, the total computational complexity of our method is $O(NK_2)+O(NK_3)=(NK_2)\,(K_3 < K_2)$. In addition, the neighborhood points can be achieved through the KD tree, and the K_2 value is much smaller than N. We use MATLAB to implement the algorithm. The implementation takes less than 0.1 s to complete keypoint detection for a 3D point cloud in a 3.7 GHz com-

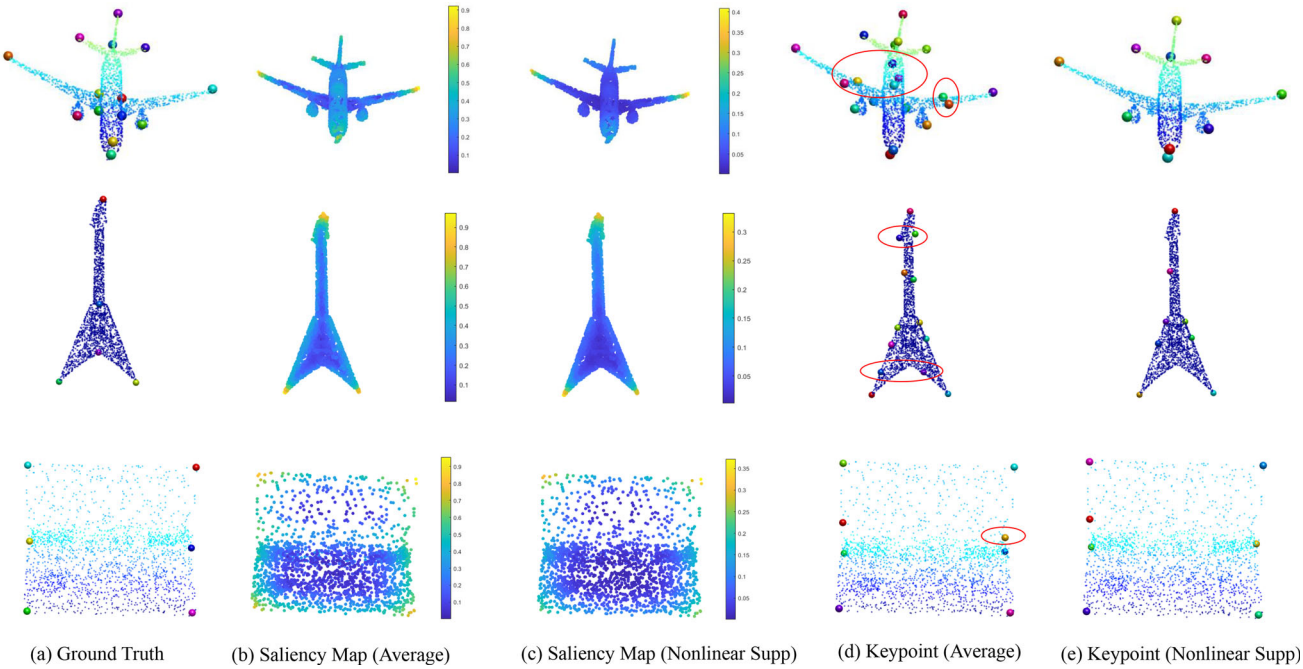


Fig. 18 (**a**) groundtruth; (**b**) and (**c**) represent the saliency maps obatained using the average aggregation and nonlinear suppression aggregation methods, respectively. The erroneous keypoint results in (**d**), marked with red ellipses using the average aggregation method, can be suppressed by the nonlinear aggregation method in (**e**)

Table 12 Comparison results of computational complexity between the FL3K method and some other approaches

Method	Computational complexity	FLOPs(G)
ISS (Zhong, 2009)	$O(DKN)$	1.18×10^{-3}
SIFT-3D (Rister et al., 2017)	$O(TSKN)$	3.15×10^{-3}
Harris-3D (Sipiran & Bustos, 2011)	$O(DK^2N)$	7.56×10^{-2}
USIP (Li & Lee, 2019)	$O(MP_1C_1KN)$	3.8
UKPGAN (You et al., 2022)	$O(NP_2WHDP_2C_2)$	85
SNAKE (Zhong et al., 2022)	$O(NP_3C_3WHDP_3C_3)$	1.24×10^3
FL3K	$O(NK_2)$	0.52×10^{-3}

puter. Therefore, our FL3K method is fast in the keypoint detection process.

To further demonstrate the efficiency of our method on keypoint detection tasks, we compare the computational complexity of our method with other methods. Table 12 presents the results of comparing the computational complexity of all the methods. We calculate the complexity of these methods according to the methods in the literature. In Table 12. N is the number of point clouds, K is the number of neighborhood points with a spherical radius r, and K_2 is the number of points in a large neighborhood R for the FL3K method. All methods other than ours use the same neighborhood size. The FL3K method uses larger neighborhood values, usually four times those of the other methods. For the ISS and Harris-3D methods, D is the dimension used to calculate the covariance matrix, T is the number of octaves to compute in SIFT-3D, and S denotes the number of scales within each octave. In deep learning-based keypoint detection methods, M represents the number of farthest point samples in the USIP method, and P_1, P_2, and P_3 represent the number of neurons in each layer of the USIP, UKPGAN, and SNAKE methods, respectively. C_1, C_2, and C_3 represent the number of network layers used by the USIP, UKPGAN, and SNAKE methods. W, H, and D represent the number of voxels in the UKPGAN and SNAKE methods. We give the O representation of computational complexity and the floating-point operations (FLOPs) of these methods. These metrics provide a better measure of the performance of these algorithms.

In calculating FLOPs, we take the same values for all the parameters except those specific to a particular method. We take the values from the original papers for specific parameters using different methods. We use $N = 2048$, $K = 64$, and $K_2 = 256$. The covariance D in ISS and Harris-3D is nine. There are six octaves, T, and four scales, S, in the SIFT-3D method. For the USIP method, $M = 64$ represents the number of points obtained by the farthest point sampling method, and $P_1 = 64$ is the number of neurons for each layer of the feature proposal network (FPN). We only use the minimum number of neurons here to simplify the calculation process for deep learning-based keypoint detectors. Deep learning methods are parameter-dependent, requiring considerable data for training and multi-layer network structures. Therefore, deep learning keypoint detectors' computational complexity and FLOPs are much higher than conventional keypoint detectors. The other values are $W = H = D = 16$, $C_1 = 7$, $C_2 = 10$, $C_3 = 6$, $P_2 = 32$, and $P_3 = 64$. These parameters are set according to the corresponding original paper. We calculate the FLOPs for these different methods based on these values. FL3K has the lowest computational complexity and FLOPS of all methods. However, deep learning-based keypoint detection methods have higher computational complexity and FLOPs than traditional ones. Because deep learning methods are data-driven and require a large amount of data for parameter learning, deep learning methods are not lightweight. Therefore, our method is fast and efficient for keypoint detection tasks and can be used for real-time analysis and processing of large-scale point clouds.

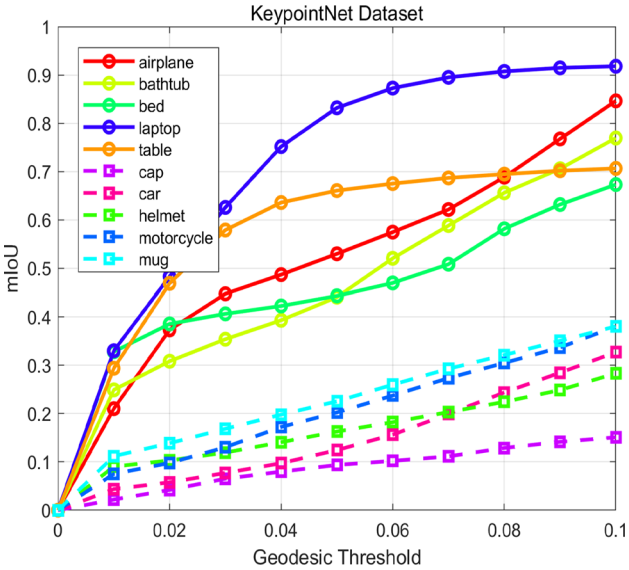
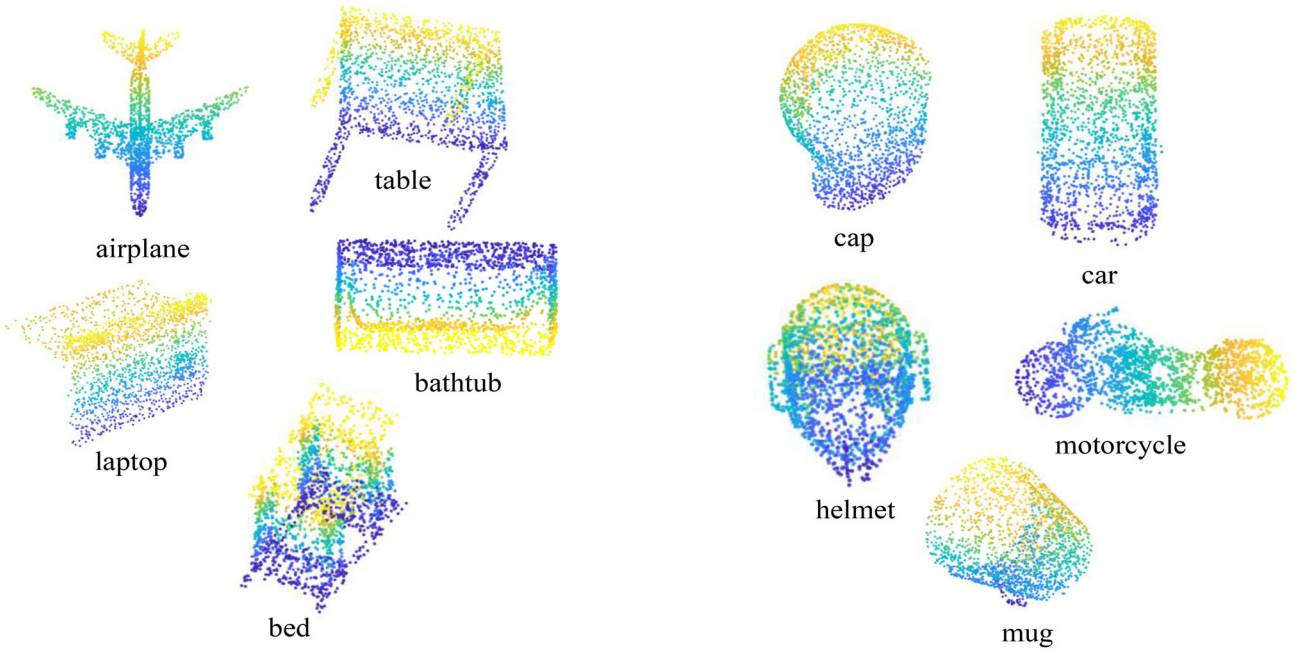


Fig. 19 Keypoint detection results of the FL3K method on ten categories of the KeypointNet dataset

4.10 Limitations

The detection accuracy of our method decreases significantly at large downsampling rates, as shown in Fig. 10b. Our future



(a) The top 5 classes with the best performance

(b) The top 5 classes with the worst performance

Fig. 20 An example of ten categories of the KeypointNet dataset. The five categories with the best performance in **a** have rich geometric features, such as more corners, and those with the worst performance in **b** have fewer geometric features

work will focus on enhancing model performance for large downsampling rates.

The FL3K method performs poorly on some categories with insignificant geometric features. For example, our method achieves the lowest performance on the five categories of caps, helmets, cars, mugs, and motorcycles in the KeypointNet dataset. In contrast, FL3K achieves the highest performance on airplanes, laptops, beds, bathtubs, and tables for the KeypointNet dataset. The keypoint detection results of our method on these ten categories are shown in Fig. 19. In addition, we present an example of these ten different categories as shown in Fig. 20. We note both geometric and regional saliency depend on the structural properties of objects. Improving the performance of keypoint detection for geometrically inexpensive object categories will be part of our future work. Introducing topological structure information of an object is a promising direction for keypoint detection.

5 Conclusion

This paper presents FL3K, a fast and lightweight 3D keypoint detection method. Its key contributions are geometric and regional saliency, which can efficiently capture a point cloud's geometric structural information and semantic information. We conduct extensive testing on benchmark 3D keypoint datasets, and the results demonstrate that our method outperforms existing handcrafted and deep learning methods. Our FL3K detector can generate stable keypoints on both rigid and nonrigid 3D objects and does not require a complex training process. It has excellent generalization. The drawback of our method is that the detection results could be more stable at large point cloud downsampling rates. In future work, we will further improve the robustness of our method under various disturbances.

Acknowledgements This work was supported by the National Natural Science Foundation of China (Grant Nos. 62106227, 62272419, 62402449, 61902159). the Teacher Professional Development Project for Domestic Visiting Scholars in 2023 (Project No. FX2023007), the National Natural Science Foundation of China Joint Fund for Regional Innovation and Development Key Support Projects (Project No. U22A20102), the Zhejiang Province Vanguard Leading Goose R&D Key Project No. 2023C01150), the Zhejiang Provincial Natural Science Foundation of China (Project No. LZ22F020010), the Open Project Program of the State Key Laboratory of CAD&CG (Grant No. A2413), Zhejiang University, and the China Postdoctoral Science Foundation (Project No. 2023M743132). We thank LetPub (www.letpub.com) and professor Daniel Morris from Michigan State University for their linguistic assistance while preparing this manuscript.

References

Bai, X., Luo, Z., Zhou, L., Fu, H., Quan, L., & Tai, C. L. (2020). D3Feat: Joint learning of dense detection and description of 3D local features. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 6359–6367).

Bai, Y., Wang, A., Kortylewski, A., & Yuille, A. (2023). CoKe: Contrastive learning for robust keypoint detection. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision* (pp. 65–74).

Barroso-Laguna, A., & Mikolajczyk, K. (2022). Key.Net: Keypoint detection by handcrafted and learned cnn filters revisited. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 45*(1), 698–711.

Boroson, E. R., & Ayanian, N. (2019). 3D keypoint repeatability for heterogeneous multi-robot SLAM. In *International conference on robotics and automation (ICRA)* (pp. 6337–6343). IEEE.

Castellani, U., Cristani, M., Fantoni, S., & Murino, V. (2008). Sparse points matching by combining 3d mesh saliency with statistical descriptors. *Computer Graphics Forum, 27*, 643–652.

Chen, H., & Bhanu, B. (2007). 3D free-form object recognition in range images using local surface patches. *Pattern Recognition Letters, 28*(10), 1252–1262.

Choi, S., Zhou, Q. Y., & Koltun, V. (2015). Robust reconstruction of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5556–5565).

Deng, X., Zuo, D., Zhang, Y., Cui, Z., Cheng, J., Tan, P., Chang, L., Pollefeys, M., Fanello, S., & Wang, H. (2022). Recurrent 3D hand pose estimation using cascaded pose-guided 3D alignments. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 45*(1), 932–945.

Gao, Y., He, J., Zhang, T., Zhang, Z., & Zhang, Y. (2023). Dynamic keypoint detection network for image matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 45*, 14404–14419.

Geng, Z., Sun, K., Xiao, B., Zhang, Z., & Wang, J. (2021). Bottom-up human pose estimation via disentangled keypoint regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 14676–14686).

Grill-Spector, K., & Malach, R. (2004). The human visual cortex. *Annual Review of Neuroscience, 27*, 649–677.

Hosmer, D. W., Hosmer, T., Le Cessie, S., & Lemeshow, S. (1997). A comparison of goodness-of-fit tests for the logistic regression model. *Statistics inMmedicine, 16*(9), 965–980.

Hu, J., Mao, M., Bao, H., Zhang, G., & Cui, Z. (2024). CP-SLAM: Collaborative neural point-based SLAM system. *Advances in Neural Information Processing Systems, 36*.

Jelavic, E., Nubert, J., & Hutter, M. (2022). Open3D SLAM: Point cloud based mapping and localization for education. In *Robotic perception and mapping: Emerging techniques, ICRA Workshop* (p. 24). ETH Zurich, Robotic Systems Lab.

Lafferty, J., McCallum, A., & Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data[C]// The Eighteenth International Conference on Machine Learning(ICML). 1(2),1–8.

Lee, C. H., Varshney, A., & Jacobs, D. W. (2005). Mesh saliency. In *ACM SIGGRAPH Papers* (pp. 659–666).

Li, J., & Lee, G. H. (2019). USIP: Unsupervised stable interest point detection from 3D point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 361–370).

Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., & Black, M. J. (2015). SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics, 34*(6), 248.

Lu, C., & Koniusz, P. (2022). Few-shot keypoint detection with uncertainty learning for unseen species. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 19416–19426).

Lu, F., Chen, G., Liu, Y., Qu, Z., & Knoll, A. (2020). RSKDD-Net: Random sample-based keypoint detector and descriptor. *Advances in Neural Information Processing Systems, 33*, 21297–21308.

Luo, Z., Xue, W., Chae, J., & Fu, G. (2022). SKP: Semantic 3D keypoint detection for category-level robotic manipulation. *IEEE Robotics and Automation Letters, 7*(2), 5437–5444.

Pomerleau, F., Liu, M., Colas, F., & Siegwart, R. (2012). Challenging data sets for point cloud registration algorithms. *The International Journal of Robotics Research, 31*(14), 1705–1711.

Prakhya, S. M., Liu, B., & Lin, W. (2016). Detecting keypoint sets on 3d point clouds via histogram of normal orientations. *Pattern Recognition Letters, 83*, 42–48.

Rister, B., Horowitz, M. A., & Rubin, D. L. (2017). Volumetric image registration from invariant keypoints. *IEEE Transactions on Image Processing, 26*(10), 4900–4910.

Shi, C., Chen, X., Huang, K., Xiao, J., Lu, H., & Stachniss, C. (2021). Keypoint matching for point cloud registration using multiplex dynamic graph attention networks. *IEEE Robotics and Automation Letters, 6*(4), 8221–8228.

Sipiran, I., & Bustos, B. (2011). Harris 3D: A robust extension of the Harris operator for interest point detection on 3d meshes. *The Visual Computer, 27*, 963–976.

Sun, J., Ovsjanikov, M., & Guibas, L. (2009). A concise and provably informative multi-scale signature based on heat diffusion. *Computer Graphics Forum, 28*, 1383–1392.

Taylor, Z. (2023). Find 3D normals and curvature. https://ww2.mathworks.cn/matlabcentral/fileexchange/48111-find-3d-normals-and-curvature

Teng, H., Chatziparaschis, D., Kan, X., Roy-Chowdhury, A. K., & Karydis, K. (2023). Centroid distance keypoint detector for colored point clouds. In *Proceedings of the IEEE/CVF Winter conference on applications of computer vision* (pp. 1196–1205).

Tinchev, G., Penate-Sanchez, A., & Fallon, M. (2021). Skd: Keypoint detection for point clouds using saliency estimation. *IEEE Robotics and Automation Letters, 6*(2), 3785–3792.

Tombari, F., Salti, S., & Di Stefano, L. (2013). Performance evaluation of 3d keypoint detectors. *International Journal of Computer Vision, 102*(1–3), 198–220.

Unnikrishnan, R., & Hebert, M. (2008). Multi-scale interest regions from unorganized point clouds. In *IEEE Computer Society Conference on computer vision and pattern recognition workshops* (pp. 1–8). IEEE.

Uy, M. A., & Lee, G. H. (2018). PointNetVLAD: Deep point cloud based retrieval for large-scale place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4470–4479).

Wang, H., Guo, J., Yan, D. M., Quan, W., & Zhang, X. (2018). Learning 3D keypoint descriptors for non-rigid shape matching. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 3–19).

Wang, Y., Yan, C., Feng, Y., Du, S., Dai, Q., & Gao, Y. (2022). STORM: Structure-based overlap matching for partial point cloud registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 45*(1), 1135–1149.

Wimmer, T., Wonka, P., & Ovsjanikov, M. (2024). Back to 3D: Few-shot 3d keypoint detection with back-projected 2d features. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4154–4164).

Yang, H., & Pavone, M. (2023). Object pose estimation with statistical guarantees: Conformal keypoint detection and geometric uncertainty propagation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8947–8958).

Yang, J., Xian, K., Xiao, Y., & Cao, Z. (2017). Performance evaluation of 3D correspondence grouping algorithms. In *2017 international conference on 3D vision (3DV)* (pp. 467–476). IEEE.

Yew, Z. J., & Lee, G. H. (2018). 3DFeat-Net: Weakly supervised local 3D features for point cloud registration. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 607–623).

Yi, L., Su, H., Guo, X., & Guibas, L. J. (2017). SyncspecCNN: Synchronized spectral CNN for 3D shape segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2282–2290).

You, Y., Liu, W., Ze, Y., Li, Y. L., Wang, W., & Lu, C. (2022). UKPGAN: A general self-supervised keypoint detector. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 17042–17051).

You, Y., Lou, Y., Li, C., Cheng, Z., Li, L., Ma, L., Lu, C., & Wang, W. (2020). KeypointNet: A large-scale 3D keypoint dataset aggregated from numerous human annotations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 13647–13656).

Zaharescu, A., Boyer, E., Varanasi, K., & Horaud, R. (2009). Surface feature detection and description with applications to mesh matching. In *IEEE conference on computer vision and pattern recognition* (pp. 373–380). IEEE.

Zeng, A., Song, S., Nießner, M., Fisher, M., Xiao, J., & Funkhouser, T. (2017). 3DMatch: Learning local geometric descriptors from RGB-D reconstructions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1802–1811).

Zhang, J., Chen, Z., & Tao, D. (2021). Towards high performance human keypoint detection. *International Journal of Computer Vision, 129*(9), 2639–2662.

Zhang, R., Zhang, C., Di, Y., Manhardt, F., Liu, X., Tombari, F., & Ji, X. (2024). Kp-red: Exploiting semantic keypoints for joint 3d shape retrieval and deformation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 20540–20550).

Zheng, Q., Gong, M., You, X., & Tao, D. (2022). A unified B-Spline framework for scale-invariant keypoint detection. *International Journal of Computer Vision, 130*(3), 777–799.

Zheng, T., Chen, C., Yuan, J., Li, B., & Ren, K. (2019). Pointcloud saliency maps. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 1598–1606).

Zhong, Y. (2009). Intrinsic shape signatures: A shape descriptor for 3D object recognition. In *IEEE 12th international conference on computer vision workshops, ICCV Workshops* (pp. 689–696). IEEE.

Zhong, C., You, P., Chen, X., Zhao, H., Sun, F., Zhou, G., Mu, X., Gan, C., & Huang, W. (2022). SNAKE: Shape-aware neural 3d keypoint field. *Advances in Neural Information Processing Systems, 35*, 7052–7064.

Zhong, C., Zheng, Y., Zheng, Y., Zhao, H., Yi, L., Mu, X., Wang, L., Li, P., Zhou, G., Yang, C., Zhang, X., & Zhao, J. (2023). 3D implicit transporter for temporally consistent keypoint discovery. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 3869–3880).

Zohaib, M., & Del Bue, A. (2023). SC3K: Self-supervised and coherent 3D keypoints estimation from rotated, noisy, and decimated point cloud data. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 22509–22519).