

Article

LRF-Net: Learning Local Reference Frames for 3D Local Shape Description and Matching

Angfan Zhu ¹, Jiaqi Yang ², Weiyue Zhao ¹ and Zhiguo Cao ^{1,*}

¹ National Key Laboratory of Science and Technology on Multi-Spectral Information Processing, School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China; zhuangfan@hust.edu.cn (A.Z.); zhaoweiye@hust.edu.cn (W.Z.)

² National Engineering Laboratory for Integrated aero-Space-Ground-Ocean Big Data Application Technology, School of Computer Science, Northwestern Polytechnical University, Xi'an 710129, China; jqyang@nwpu.edu.cn

* Correspondence: zgcao@hust.edu.cn; Tel.: +86-027-8755-8918; Fax: +86-027-8754-3594

Received: 29 July 2020; Accepted: 3 September 2020; Published: 7 September 2020



Abstract: The local reference frame (LRF) acts as a critical role in 3D local shape description and matching. However, most existing LRFs are hand-crafted and suffer from limited repeatability and robustness. This paper presents the first attempt to learn an LRF via a Siamese network that needs weak supervision only. In particular, we argue that each neighboring point in the local surface gives a unique contribution to LRF construction and measure such contributions via learned weights. Extensive analysis and comparative experiments on three public datasets addressing different application scenarios have demonstrated that LRF-Net is more repeatable and robust than several state-of-the-art LRF methods (LRF-Net is only trained on one dataset). We show that LRFNet achieves 0.686 MeanCos performance on the UWA 3D modeling (UWA3M) dataset, outperforming the closest method by 0.18. In addition, LRF-Net can significantly boost the local shape description and 6-DoF pose estimation performance when matching 3D point clouds.

Keywords: point cloud; local reference frame; deep learning

1. Introduction

The local reference frame (LRF) is a canonical coordinate system established in the 3D local surface, which is a useful geometric cue for 3D point clouds. LRF possesses two intriguing traits. One is that rotation invariance can be achieved via LRF if the local surface is transformed with respect to the LRF [1]. The other is that useful geometric information can be mined with LRF [2]. These make LRF popular in many geometric relevant tasks, especially for local shape description and six-degree-of-free (6-DoF) pose estimation.

For local shape description, two corresponding local surfaces can be converted into the same pose and full 3D geometric information can be employed, which is beneficial to improving the performance of local descriptors. Some hand-crafted local shape descriptors, e.g., signature of histograms of orientations (SHOT) [3] and signature of rotational projection statistics (RoPS) [1], estimate an LRF from the local surface and then translate local geometric information with respect to the estimated LRF into distinctive and rotation-invariant feature representations. Some learned local descriptors, e.g., [4,5], leverage LRFs to overcome the limitation of geometric deep learning networks of being sensitive to rotations. Therefore, LRF is critical for both traditional and learned local shape descriptors. For 6-DoF pose estimation, an LRF can significantly improve its efficiency. Traditional 6-DoF pose estimation is usually performed via RANSAC [6], which randomly selects inlier correspondences from an initial correspondence pool for pose prediction. Such random sampling method is neither

reliable nor computational efficient [7]. By contrast, we can directly predict an initial pose via two corresponding LRFs, reducing the computational complexity from $O(n^3)$ to $O(n)$.

The desirable properties for LRF are twofold [3]. The first one is the invariance to rigid transformation (e.g., translations and rotations). The second one is the robustness to common disturbances (e.g., noise, clutter, occlusion and varying mesh resolutions). To achieve these goals, many LRF methods have been proposed in the past decade and they can be categorized into two classes [8]: Covariance analysis (CA) [3,9] or point spatial distributions (PSD)-based [2,10,11]. CA-based LRFs are based on the computation of eigenvectors of a covariance matrix calculated either for the points or triangles in the local surface. PSD-based LRFs usually calculate axes successively, where the main efforts are put on the determination of the x -axis [8]. However, most CA-based LRFs still suffer from sign ambiguity, and PSD-based LRFs show limited robustness to high levels of noise and variations of mesh resolution [10]. Methods in both classes usually apply a weighted strategy to improve their repeatability performance. However, their weights are determined heuristically, and the repeatability performance in challenging 3D matching cases cannot be guaranteed.

Motivated by existing considerations, we propose a learned approach toward LRF estimation (named LRF-Net), which considers the contribution of all neighboring points (Figure 1). Our key insight is that each neighboring point in the local surface gives a unique contribution to LRF construction, which can be quantitatively represented by assigning weights to these points. Given a local surface centered at a keypoint, we first resort to the normal of the keypoint computed within a subset of the radius neighbors for the calculation of its z -axis. Its repeatability has been confirmed in [2]. Compared with z -axis, estimating the x -axis is more challenging, due to noise, clutter, and occlusion. By collecting angle and distance attributes within a local neighborhood, we can formulate the estimation of x -axis as a weighted prediction problem with respect to these geometric attributes. Note that, we choose these invariant geometric attributes instead of raw points as input to our LRF-Net. The distance and angle computation are mathematically invariant under isometric transformation and hence per definition invariant to rigid body motion. Unlike previous CA-based and PSD-based approaches, such learned strategy of determining weights is shown to be invariant to rigid transformation and robust to noise, clutter, occlusion and varying mesh resolutions. Our network can be trained in a weakly supervised manner. Specifically, it needs the corresponding relationships between local patches only, instead of ground-truth LRFs and/or exact pose variation information between patches. We have conducted a set of experiments on three public datasets to comprehensively evaluate the proposed LRF-Net. Extensive analysis and comparative experiments on three public datasets addressing different application scenarios have demonstrated that LRF-Net is more repeatable and robust than several state-of-the-art LRF methods (LRF-Net is only trained on one dataset). In addition, LRF-Net can significantly boost the local shape description and 6-DoF pose estimation performance when matching 3D point clouds. The major contributions of this paper are summarized as follows:

- LRF-Net, based on a Siamese network that needs weak supervision only, is proposed that achieves the state-of-the-art repeatability performance under the impacts of noise, varying mesh resolutions, clutter and occlusion. To the best of our knowledge, we are the first to concentrate on designing LRF for local surfaces with deep learning.
- LRF-Net can significantly boost the performance of local shape description and 6-DoF pose estimation.



Figure 1. Local reference frame (LRF)-Net first assigns learned weights to points in a local surface and then uses these weights to estimate a repeatable and robust LRF.

The rest of this paper is organized as follows. Section 2 presents a detailed description of our proposed LRF-Net. Section 3 presents the experimental evaluation of LRF-Net on three public datasets with comparisons with several state-of-the-art methods. Several concluding remarks are drawn in Section 4.

2. Related Works

Various methods for building LRFs have been proposed in the literature. Most of them can be categorized into two classes: CA-based methods and PSD-based methods. Given a local surface with a spherical support of radius r centered at the keypoint p , they compute a 3×3 matrix as its LRF.

2.1. CA-Based LRF Methods

Most CA-based methods are based on the eigenvectors of the covariance matrix, which is usually generated by the points or triangles in the support region.

Mian et al. [9]: This method directly calculates the unit vectors of the LRF via computing covariance analysis on the radius neighbors of the keypoint, the three eigenvectors of the covariance matrix are defined as the x, y, z -axis, respectively. While the eigenvectors of the covariance matrix define the principal direction of the local surface, their sign is still ambiguous [10]. Mian et al. disambiguates the sign of z -axis through the inner product between $\mathbf{n}(p)$ (normal of keypoint p) and two possible vector, i.e., $\mathbf{z}(p)$ and $-\mathbf{z}(p)$, where $\mathbf{z}(p)$ denotes the z -axis. However, the rest axes are still suffer from sign ambiguity.

SHOT [3]: This method leverages a weighted covariance matrix for the computation of LRF, which assigns smaller weights to more distant points. The weighted covariance matrix is calculated as follows:

$$\mathbf{C}_{shot} = \frac{1}{\sum_{q \in N(p)} w_q} \sum_{q \in N(p)} w_q (q - p)(q - p)^T \quad (1)$$

where $w_q = R - \|q - p\|$. R denotes the support radius and $\|\cdot\|$ represents L_2 norm. This weighted strategy improves the repeatability in present of clutter under 3D object recognition scenarios. To eliminate all sign ambiguities of the LRF axes, a technique which is similar to [12] is applied to the eigenvectors of the weighted covariance matrix. Specifically, the sign of a eigenvector is reoriented to coherent with the majority of the vectors. Such technique is used on the x -axis and z -axis. The rest y -axis is calculated by the cross-product operation between the z -axis and the x -axis.

RoPS [1]: This method does not only calculate one covariance matrix for the local surface, it aggregates multiple covariance matrices computed for every single triangle of the local surface into a comprehensive one to enhance the robustness. Such method needs mesh representation of the 3D local surface. For a triangle $\tau \in \psi(p)$, its covariance matrix is calculated as:

$$C_{\tau} = \frac{1}{12} \sum_{i=1}^3 \sum_{j=1}^3 (q_i^{\tau} - p)(q_j^{\tau} - p)^T + \frac{1}{12} \sum_{i=1}^3 (q_i^{\tau} - p)(q_i^{\tau} - p)^T \quad (2)$$

where q_1^{τ} , q_2^{τ} and q_3^{τ} denote the three vertices of τ . Then, the comprehensive covariance matrix is calculated as:

$$C_{rops} = \sum_{\tau \in \psi(p)} w_1 w_2 C_{\tau} \quad (3)$$

w_1 and w_2 are defined as:

$$w_1 = \frac{|(q_2^{\tau} - q_1^{\tau}) \times (q_3^{\tau} - q_1^{\tau})|}{\sum_{\tau \in \psi(p)} |(q_2^{\tau} - q_1^{\tau}) \times (q_3^{\tau} - q_1^{\tau})|} \quad (4)$$

$$w_2 = (R - |p - \frac{q_1^{\tau} + q_2^{\tau} + q_3^{\tau}}{3}|)^2 \quad (5)$$

where w_1 alleviates the impact of mesh resolution variations and w_2 improves the robustness performance to clutter and occlusion [8]. Based on the eigenvalue decomposition of C_{rops} , the three axes of LRF can be calculated.

As for disambiguating the sign, x -axis and z -axis (only take x -axis as an example) are further adjusted via $\mathbf{x}(p) = \mathbf{x}(p) \cdot \text{sign}(h)$, where $\mathbf{x}(p)$ denotes the x -axis and h is a signum function, which is defined as:

$$h = \sum_{\tau \in \psi(p)} w_1 w_2 \left(\frac{1}{6} \sum_{i=1}^3 (q_i^{\tau} - p) \cdot \mathbf{x}(p) \right) \quad (6)$$

Once the x -axis and z -axis are determined, the y -axis can be calculated via the cross-product between them.

2.2. PSD-Based LRF Methods

As for PSD-based LRF methods, they calculate three axes of the LRF successively.

PS [13]: This method puts a sphere of radius r on the keypoint p and gain a contour at the intersection of the local surface. The point with the biggest signed projection distance to the tangent plane of the keypoint was selected to compute the x -axis, while the tangent plane is determined by z -axis, which is directly performed by the normal of the keypoint. The y -axis is calculated via the cross-product operation.

Board [2]: This method collects a small subset of the local surface for the estimation of the z -axis, which has achieved a robust performance to occlusion. The x -axis is calculated by the points lying in the border region. They choose the point lying in the border region with the biggest deviation angle between its normal and the z -axis as the calculation of x -axis. The y -axis is computed by the cross-product operation between z -axis and x -axis.

SD [10]: This method is a modified version of Board [2]. They make improvement to the repeatability of the LRF via employing the point with largest local depth instead of deviation angle in SD [10]. They achieve a more repeatable performance than Board on 3D registration and recognition data. However, both of them show a weak performance on the robustness to the large scale noise.

TOLDI [11]: This method resorts to the normal of the keypoint which is calculated by a subset of the radius neighbors for the estimation of its z -axis. Then, the tangent plane of the keypoint with respect to z -axis is determined and all radius neighbors of the keypoint are projected on the tangent plane. A weighted strategy is employed to each projection vector to calculate the x -axis, which is defined as:

$$w_{i1} = (r - \|p - q_i\|)^2 \quad (7)$$

$$w_{i2} = (\mathbf{p} \mathbf{q}_i \cdot \mathbf{z}(\mathbf{p}))^2 \quad (8)$$

where p donates the keypoint and q_i is one of its radius neighbors within support radius r . w_{i1} is a weight related to the distance from p to q_i , which is designed to improve the robustness of the LRF to clutter, occlusion and incomplete border regions [11]. w_{i2} is a weight related to the local depth which is designed to provide high repeatability on flat regions [11]. The x -axis is calculated as:

$$\mathbf{x}(\mathbf{p}) = \sum_{i=1}^k w_{i1} w_{i2} \mathbf{v}_i / \left\| \sum_{i=1}^k w_{i1} w_{i2} \mathbf{v}_i \right\| \quad (9)$$

where k is the count of radius neighbors of keypoint p and $\mathbf{v}_i$ denotes one of the projection vectors. The y -axis is computed by the cross-product between z -axis and x -axis.

3. Methods

This section represents the details of our proposed LRF-Net for 3D local surface. We first introduce the technique approach for calculating the three axes for an LRF and then describes a weakly supervised approach for training LRF-Net.

3.1. A Learned LRF Proposal

The whole architecture of LRF-Net is shown in Figure 2a. LRF-Net predicts the direction of three axes successively. For a local surface, we first estimate its z -axis via its normal vector computed over a small subset of the local point set. Then, unique weights are learned for each point in the local surface. The x -axis is calculated by integrating projection vectors with learned weights using a vector-sum operation. At last, the y -axis is calculated by the cross-product operation between z -axis and x -axis.

LRF definition: Given a local surface $\mathbf{Q}$ centered at keypoint $\mathbf{p}$, the LRF at $\mathbf{p}$ (denoted by $\mathbf{L}_p$) can be represented as :

$$\mathbf{L}_p = [\mathbf{x}(\mathbf{p}), \mathbf{z}(\mathbf{p}) \times \mathbf{x}(\mathbf{p}), \mathbf{z}(\mathbf{p})], \quad (10)$$

where $\mathbf{x}(\mathbf{p})$, $\mathbf{y}(\mathbf{p})$, and $\mathbf{z}(\mathbf{p})$ denote the x -axis, y -axis, and z -axis of $\mathbf{L}_p$, respectively. As three axes are orthogonal, the estimation of LRF therefore contains two parts: Estimation of the z -axis and the x -axis.

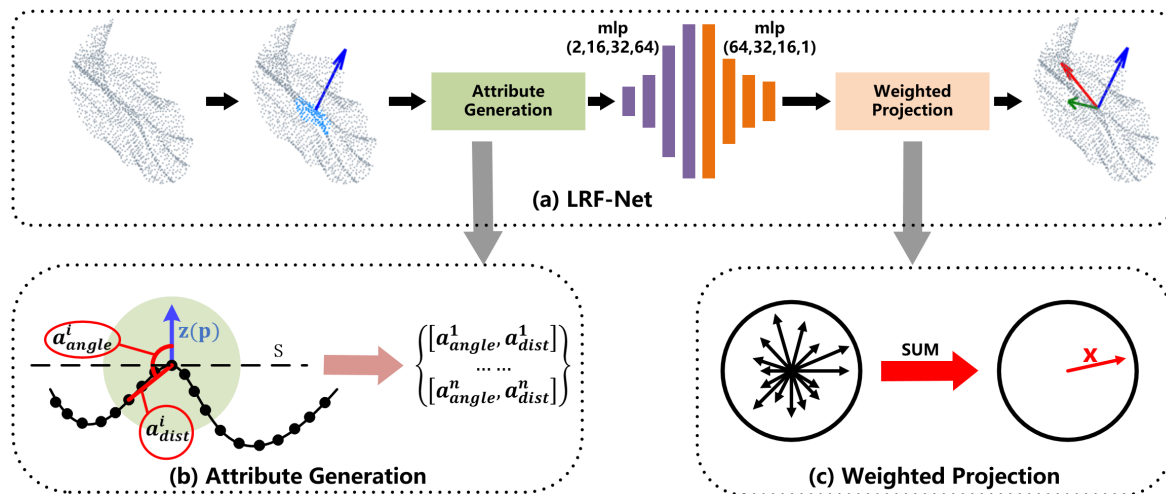


Figure 2. The architecture of LRF-Net. The input to LRF-Net is a local surface and we calculate its normal as the z -axis of the LRF. Then, the local surface is converted to a set of rotation variant attributes. Next, a projection weight for every point is computed with mlp. At last, the x -axis is calculated by the weighted vector-sum of all the projection vectors and the y -axis is calculated by the cross product between the z -axis and x -axis. The LRF is formed as the combination of the x -axis, y -axis and z -axis.

A naive way to learn an LRF for the local surface is to train a network that directly regresses the axes. The premise is that ground-truth LRFs are labeled for local surfaces. Unfortunately, the network trained in this manner meets two difficulties. The first one is that the definition of ground-truth LRFs for local surfaces remain an open issue in the community [8]. The second one, which is more important, is that the orthogonality of three axes cannot be guaranteed. We suggest estimating z -axis and x -axis independently.

Z-axis: As for z -axis, we take the normal of the keypoint as the z -axis., which has been confirmed [2] to be quite repeatable. To resist the impact of clutter and occlusion, we collect a small subset of the local surface to calculate the normal. For more details, readers are referred to [11].

X-axis: Once the z -axis is determined, the remaining task is to compute the x -axis. Compared with z -axis, x -axis is more challenging due to the influence of noise, clutter, and occlusion [8]. We argue that each neighboring point in the local surface gives a unique contribution to LRF construction. Hence, we predict a weight for each neighboring point and leverage all neighboring points with learned weights for x -axis prediction. The main steps are as follows.

First, to make the estimate LRF invariant to rigid transformation, our network consumes with invariant geometric attributes, rather than point coordinates. In particular, two attributes, i.e., relative distance a_{dist} and surface variation angle a_{angle} are used in LRF-Net as illustrated in Figure 2b. For a neighbor $\mathbf{q}_i$ of $\mathbf{p}$, the two attributes of $\mathbf{q}_i$ are computed as:

$$\begin{cases} a_{dist}^i = \|\mathbf{p}\mathbf{q}_i\| / r \\ a_{angle}^i = \cos(\mathbf{z}(\mathbf{p}), \mathbf{p}\mathbf{q}_i) \end{cases} \quad (11)$$

where $\|\cdot\|$ is the L_2 norm and r represents the support radius of the local surface. The range of a_{angle} and a_{dist} are $[-1, 1]$ and $[0, 1]$, respectively. Thus, every radius neighboring point represented by two attributes that will be encoded to a weight value via LRF-Net later. The employed two attributes in LRF-Net have two merits at least. First, the unique spatial information of a radius neighboring point in the local surface can be well represented, as shown in Figure 3. Both attributes are complementary to each other. Second, the two attributes are calculated with respect to the keypoint, which are rotation invariant. It makes the learned weights rotation invariant as well.

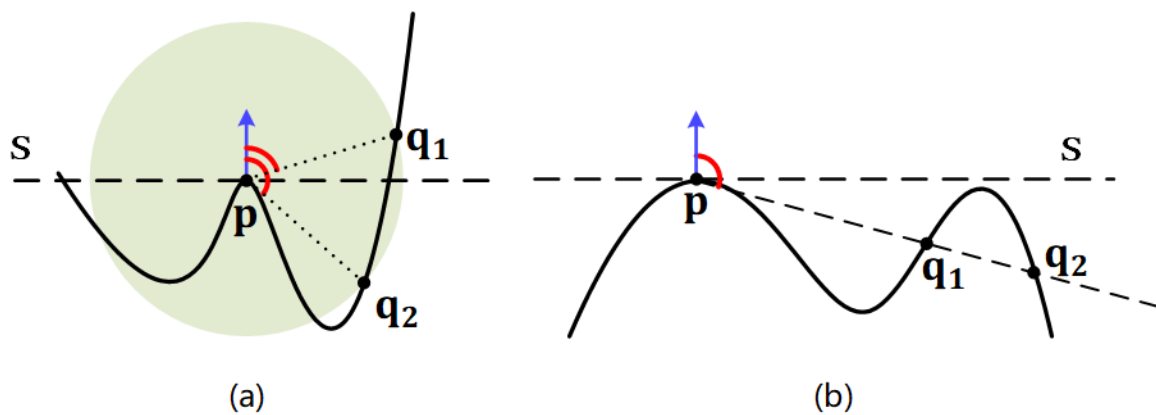


Figure 3. An illustration of information complementary inherent to the two attributes in LRF-Net. The two radius neighbors $\mathbf{q}_1$ and $\mathbf{q}_2$ of the keypoint $\mathbf{p}$ in (a) and (b) have different spatial locations. In (a), the two radius neighbors with the same distance value are distinguished by the surface variation angle attribute. In (b), their surface variation angle attribute values are similar, while they can be distinguished by the distance attribute.

Second, with geometric attributes being the input, we use a simple network with multilayer perceptions (MLP) layers only to predict weights for neighboring points. The details of the network are illustrated in Figure 4. The network is very simple, however, is sufficient to predict stable and informative weights for neighboring points (as will be verified in the experiments).

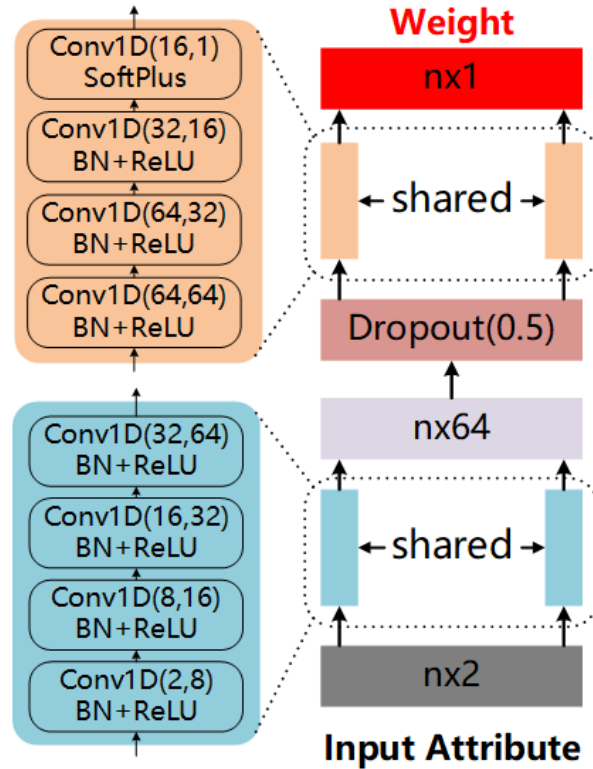


Figure 4. Parameters of our LRF-Net.

Third, because x -axis is orthogonal to z -axis, we project each neighbor $\mathbf{q}_i$ on the tangent plane $\mathbf{S}$ of the z -axis and compute a projection vector for $\mathbf{q}_i$ as:

$$\mathbf{v}_i = \mathbf{p}\mathbf{q}_i - (\mathbf{p}\mathbf{q}_i \cdot \mathbf{z}(\mathbf{p})) \cdot \mathbf{z}(\mathbf{p}). \quad (12)$$

We integrate all weighted projection vectors in a weighted vector-sum manner:

$$\mathbf{x}(\mathbf{p}) = \sum_{i=1}^n w_i \mathbf{v}_i / \left\| \sum_{i=1}^n w_i \mathbf{v}_i \right\|, \quad (13)$$

where n denotes the total number of radius neighbors of keypoint $\mathbf{p}$ and w_i is a learned weight by LRF-Net. Another way for determining the x -axis, based on these weights, is choosing the vector with the maximum weight, as in many PSD-based LRFs [2,10]. However, it fails to leverage all neighboring information and we will shown that it is inferior to the vector-sum operation in the experiments.

Y -axis: Based on the calculated z -axis and x -axis, the y -axis can be computed by the cross-product between them.

3.2. Weakly Supervised Training Scheme

Our training data are constituted by a series of corresponding local surface patches. The corresponding relationship is obtained based on the ground-truth rigid transformation of two whole point clouds. In particular, LRF-Net needs the corresponding relationships between local surface

patches only, rather than ground-truth LRFs and/or exact pose variation information between patches. Therefore, our network can be trained in a weakly supervised manner.

We train our LRF-Net with two streams in a Siamese fashion where each stream independently predicts an LRF for a local surface. Specifically, two streams take the local surfaces of keypoints $\mathbf{p}_m$ and $\mathbf{p}_s$ as inputs, respectively. Here, $\mathbf{p}_m$ and $\mathbf{p}_s$ are two corresponding keypoints sampled from the model and scene point cloud. Both streams share the same architecture and underlying weights. We use the predicted LRFs $\mathbf{L}_m$ and $\mathbf{L}_s$ by two stream to transform the local surfaces $\mathbf{Q}_m$ and $\mathbf{Q}_s$ to the coordinate system of the two LRFs. Then, we calculate the Chamfer Distance [14] between two transformed local surfaces as the loss function to train LRF-Net:

$$Loss = d_{cham}(\mathbf{L}_m \cdot \mathbf{Q}_m, \mathbf{L}_s \cdot \mathbf{Q}_s), \quad (14)$$

where

$$d_{cham}(X, \hat{X}) = \min \left\{ \frac{1}{|X|} \sum_{x \in X} \min_{\hat{x} \in \hat{X}} \|x - \hat{x}\|, \frac{1}{|\hat{X}|} \sum_{\hat{x} \in \hat{X}} \min_{x \in X} \|\hat{x} - x\| \right\}. \quad (15)$$

Our opinion is that it is difficult to define a “good” LRF for a single local surface. For 3D shape matching, LRFs that can align the poses of two local surface patches are judged as repeatable. This motivates us to consider two local patches simultaneously and employ the Chamfer Distance to train the network.

4. Experiments

In this section, we first evaluate the repeatability performance of our LRF-Net on three standard datasets, including the Bologna retrieval (BR) dataset [15], the UWA 3D modeling (UWA3M) dataset [16], and the UWA object recognition (UWAOR) dataset [17], together with a comparison with other state-of-the-art LRFs. Second, we apply our LRF-Net perform local shape description and 6-DoF pose estimation to verify the practicability of our method. Third, analysis experiments are conducted to improve the explainability of the proposed LRF-Net.

4.1. Experimental Setup

The details of our experiments including the description of datasets and the illustration for all compared methods are introduced before evaluation. The experiments were conducted on a Windows Server with an Intel Xeon E5-2640 2.39 GHz CPU and 96 GB of RAM. We train our LRF-Net using a batch size of 512 local surface pairs with Pytorch and leverage the ADAM optimizer with an initial learning rate of 1×10^{-4} , which decays 5% every epoch. Each sampled local surface contains 256 points. The max epoch count is set to 20.

4.1.1. Datasets

Our experimental datasets includes three standard datasets with different application scenarios. The variety among these public 3D datasets definitely helps us to evaluate the performance of our method in a comprehensive manner. Figure 5 displays two exemplar models and scenes without noise in each dataset. The main properties of these datasets are summarized in Table 1.

These dataset are also injected with five levels of Gaussian noise (i.e., from 0.1 mr to 0.5 mr Gaussian noise) and four levels of mesh decimation (i.e., $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$ and $\frac{1}{16}$ of original mesh resolution). Here, the unit mr denotes mesh resolution. Remarkably, the noise-free BR dataset is used to train our LRF-Net, the rest noisy data in the BR dataset and data in the UWA3M dataset and the UWAOR dataset are used for testing.

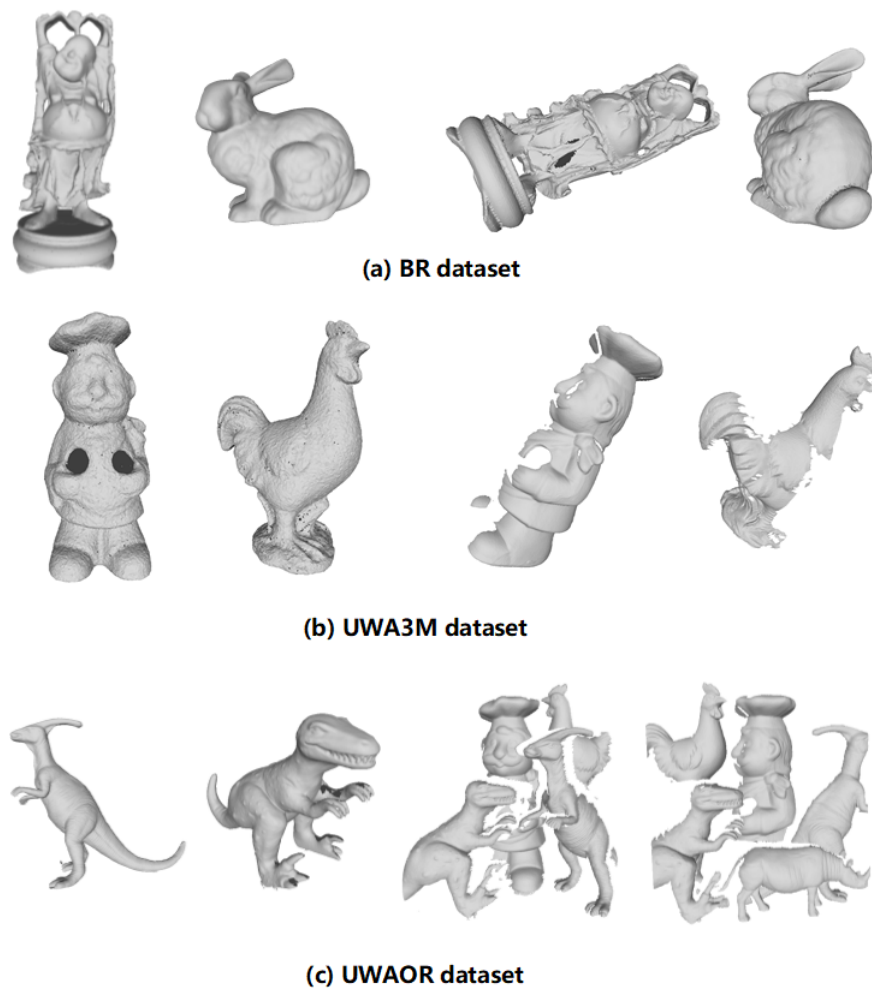


Figure 5. Two exemplar models and scenes without noise (shown from left to right), respectively, taken from the BR, UWA3M, and UWAOR datasets.

Table 1. Experimental datasets and inherited properties.

Dateset	BR	UWA3M	UWAOR
Scenario	Retrieval	Registration	Recognition
Challenge	Gaussian noise	holes, missing region, and self-occlusion	clutter and occlusion
# Models	6	4	5
# Scenes	18	75	50
# Matching Pairs	18	75	188

means the quantitative attribute of the dataset (e.g., the number of models).

4.1.2. Compared Methods

We compare our LRF-Net with several existing LRF methods for a thorough evaluation. Specifically, the compared methods are proposed by Mian et al. [9], Tombari et al. [3], Petrelli et al. [10], Guo et al. [1] and Yang et al. [11], respectively. We dub them as *Mian*, *Tombari*, *Petrelli*, *Guo*, and *Yang*, respectively. To compare fairly, we keep the support radius of all the LRFs as 15 mr. The properties of these LRFs are shown in Table 2.

To evaluate the local shape description performance of our method, we replace the LRF in four LRF-based descriptors (i.e., snapshots [18], SHOT [3], RoPS [1] and TOLDI [11]) and assess the performance variations. To measure the 6-DoF pose estimation performance of our method, we adapt LRF-Net to the RANSAC pipeline and compare with the original RANSAC [19].

Table 2. Properties of six LRF methods. H and L, respectively, represent hand-crafted and learned methods for point weight calculation; P and M, respectively, denote point cloud and mesh.

Method	Mian	Tombari	Guo	Petrelli	Yang	Ours
Category	CA	CA	CA	PSD	PSD	PSD
Date type	P	P	M	P	P	P
Weight	—	H	H	H	H	L

4.2. Performance Evaluation of LRF-Net

4.2.1. Repeatability Performance

We evaluate the repeatability of all LRFs via the popular MeanCos [3] metric, which measures overall angular error between two LRFs. The MeanCos criterion is computed as:

$$\text{MeanCos}(\mathbf{L}_m, \mathbf{L}'_s) = \frac{\text{Cos}(X) + \text{Cos}(Z)}{2} \quad (16)$$

$$\mathbf{L}'_s = \mathbf{L}_s * \mathbf{GT} \quad (17)$$

where $\mathbf{L}_m$ and $\mathbf{L}_s$ denote two corresponding LRFs between model and scene. $\mathbf{L}'_s$ represents the transformed $\mathbf{L}_s$, gained via ground truth transformation $\mathbf{GT}$. $*$ denotes matrix-product. $\text{Cos}(Z)$ represents the cosine of the angle between the z-axis of the $\mathbf{L}_m$ and the $\mathbf{L}'_s$, and $\text{Cos}(X)$ coincides with the x-axis angular error between $\mathbf{L}_m$ and the $\mathbf{L}'_s$. Due to the y-axis can always be calculated from the other two axes via cross-product, it is not necessary to be included in MeanCos calculation [2]. In our evaluation, we first randomly select 1000 points from each models and collect the corresponding points in the scenes via ground truth transformation for each model-scene pair. Then, we calculate the LRF for every local surface centered at the selected point in the model and scene. At last, the average MeanCos of the MeanCos value of all the corresponding LRFs between each model-scene pair is calculated as the final result for a dataset. Note that, the MeanCos of two perfectly corresponding LRFs equals to 1. The repeatability results of evaluated LRFs are shown in Figures 6 and 7. Several observations can be made from these figures.

First, as witnessed by Figure 6, our LRF together with *Tombari*, *Petrelli*, and *Yang* achieve decent performance on the BR dataset. On the UWA3M and UWAOR datasets, our LRF-Net achieves the best performance. Second, as shown in Figure 7a, LRF-Net and *Tombari* achieve a comparably stable performance on the BR dataset with respect to different levels of Gaussian noise. Figure 7b,c indicate that LRF-Net achieves the best performance under all levels of Gaussian noise on the UWA3M and UWAOR datasets, surpassing the others by a very significant gap. Note that UWA3M and UWAOR datasets also include nuisances such as clutter, self-occlusion, and occlusion. Third, results in Figure 7d–f suggest that LRF-Net is the best competitor with $\frac{1}{2}$, $\frac{1}{4}$, and $\frac{1}{8}$ mesh decimation on all datasets.

These results clearly demonstrate the strong robustness of our LRF-Net with respect to Gaussian noise, mesh decimation, clutter, and occlusion. The reasons are at least twofold. One is that all points are leveraged to generate the critical x-axis, which guarantees the robustness to Gaussian noise and low level mesh decimation. The other is that a LRF-Net can learn stable and informative weights for neighboring points. It can improve the robustness of LRF-Net to common nuisances.

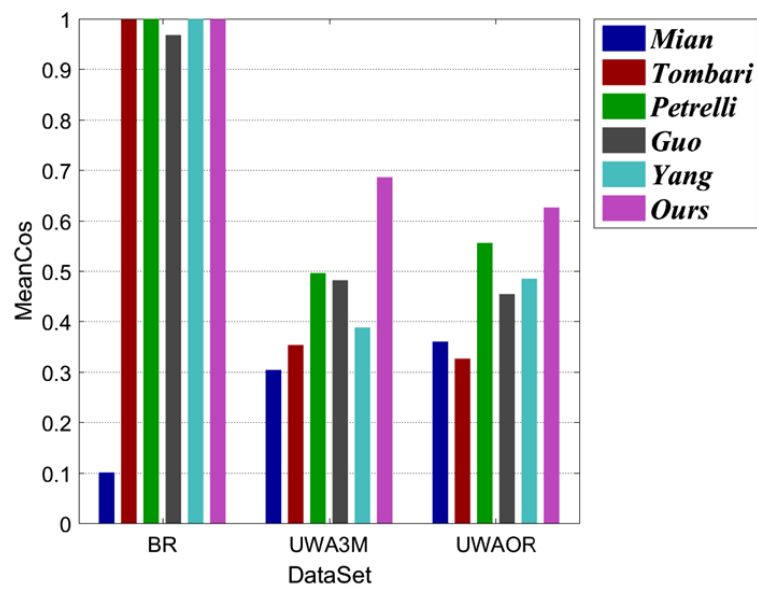


Figure 6. Repeatability performance of six LRF methods on the Bologna retrieval (BR), UWA 3D modeling (UWA3M), and UWA object recognition (UWAOR) datasets.

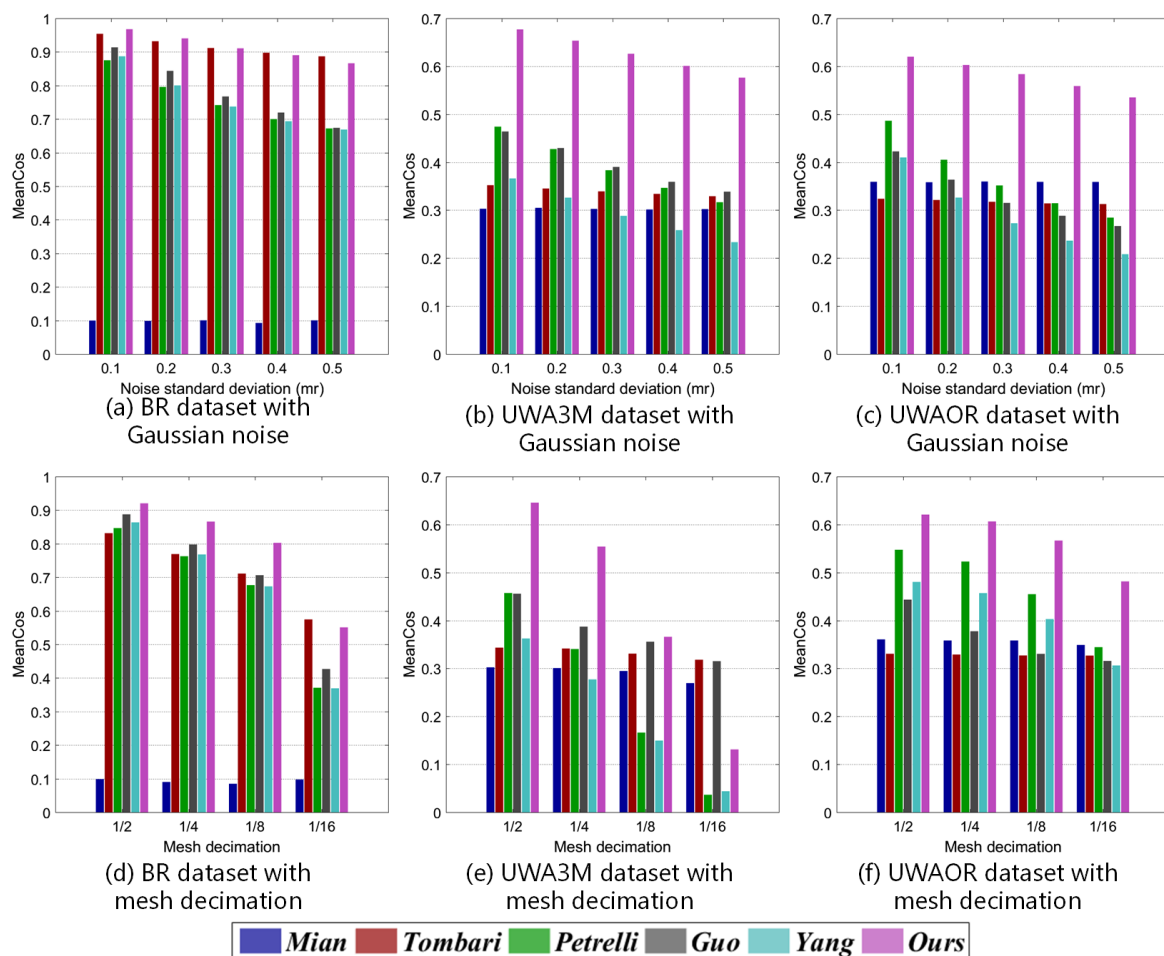


Figure 7. Robustness performance of six LRF methods on the BR, UWA3M, and UWAOR datasets with Gaussian noise and mesh decimation.

4.2.2. Local Shape Description Performance

We further evaluate our LRF-Net by replacing the LRFs in four LRF-based descriptors (i.e., snapshots, SHOT, RoPS, and TOLDI) with our LRF-Net. Then we compare their descriptor matching performance measured via recall vs. 1-precision curve (RPC) [3,20]. The calculation of recall is defined as:

$$recall = \frac{N_{true}}{N_{corr}} \quad (18)$$

where N_{true} denotes the number of correct matches and N_{corr} is the total number of corresponding features. The calculation of 1-precision is defined as:

$$1 - precision = \frac{N_{false}}{N_{match}} \quad (19)$$

where N_{false} represents the number of false matches and N_{match} is the total number of matches.

Notably, the original LRF methods employed by snapshots, SHOT, RoPS, and TOLDI are *Mian*, *Tombari*, *Guo Yang*, respectively. We conduct this experiment on the original BR, UWA3M, and UWAOR datasets. Figure 8 reports the RPC results of the all tested descriptors.

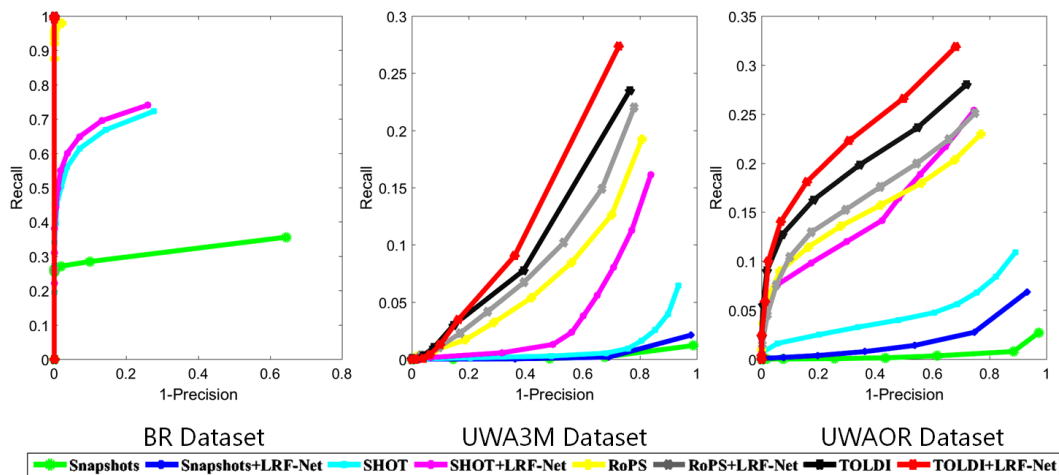


Figure 8. Local shape description performance of LRF-based descriptors with LRF-Net and their original LRFs on the BR, UWA3M, and UWAOR datasets.

As witnessed by Figure 8 and Table 3, most LRF-based descriptors equipped with our LRF-Net outperform their original versions. Specifically, snapshots achieves a dramatic performance improvement with our LRF-Net on the BR dataset; the performance of SHOT also climbs significantly on the UWA3M and UWAOR datasets with the help of the proposed LRF-Net. Therefore, we can draw a conclusion that LRF plays an important role in local shape description, where a repeatable LRF can effectively improve the description performance of an LRF-based descriptor without changing its feature representation. It also indicates that the proposed LRF-Net can bring positive impacts on a number of existing local shape descriptors.

Table 3. Overall accuracy of eight LRF-based descriptors with LRF-Net (denoted by L) and their original LRFs on the BR, UWA3M, and UWAOR datasets.

	Snapshots	Snapshots+L	SHOT	SHOT+L	RoPS	RoPS+L	TOLDI	TOLDI+L
BR	0.3733	0.9066	0.5489	0.5801	0.8827	0.9462	0.9084	0.9082
UWA3M	0.0016	0.0023	0.0209	0.0625	0.0661	0.0787	0.0417	0.0484
UWAOR	0.0048	0.0158	0.0672	0.0672	0.1623	0.1806	0.1558	0.1721

The bold means the better performance compared with other methods.

4.2.3. 6-DoF Pose Estimation Performance

A general 6-DoF pose estimation process with local descriptors is achieved by correspondence generation and pose estimation from correspondences with potential outliers [6]. RANSAC is arguably the de facto 6-DoF pose estimator in many applications. However, a key limitation of RANSAC is that the computational complexity of RANSAC is $O(n^3)$ and estimating a reasonable pose requires a huge number of iterations. With LRFs, a single correspondence is able to generate a 6-DoF pose (shown as Figure 9), decreasing the computational complexity from $O(n^3)$ to $O(n)$. Therefore, we apply LRF-Net to 6-DoF pose estimation, following a RANSAC-fashion pipeline. The difference is that we sample one correspondence per iteration. Two criteria, i.e., the rotation error err_r between our predicted rotation R and the ground-truth one R_{GT} , and the translation error err_t between the predicted translation vector T and the ground truth one T_{GT} [16], are employed for evaluating the performance of 6-DoF pose estimation. err_r and err_t are defined as:

$$err_r = \arccos\left(\frac{\text{trace}(R' - 1)}{2}\right) \frac{180}{\pi} \quad (20)$$

$$err_t = \frac{\|T_{GT} - T\|}{mr} \quad (21)$$

where $R' = R_{GT}(R)^{-1}$ and mr denotes the mesh resolution.

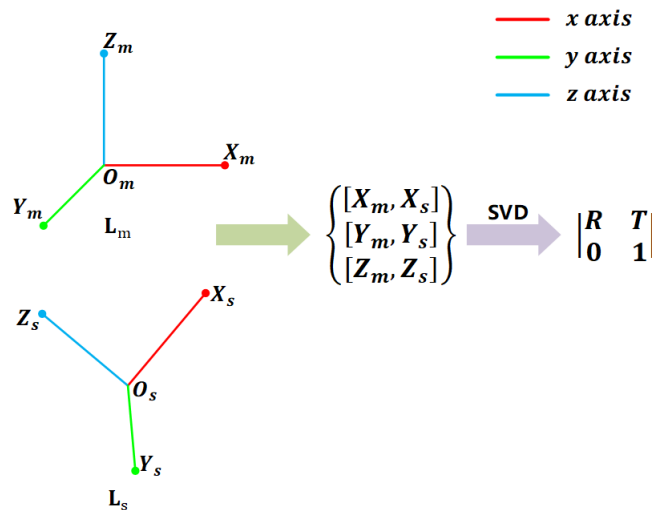


Figure 9. Illustration of directly calculating an initial pose via a single correspondence. We generate three corresponding point pairs via the centroids and LRFs of the corresponding local surface pair. The final pose is computed via SVD, which is an inner function in PCL [21].

The initial feature correspondence set is generated by first matching TOLDI (equipped with our LRF-Net) descriptors and keeping 100 correspondences with the highest similarity scores. 100 and 1000 iterations are assigned to our method and RANSAC. The average rotation errors and translation errors of the two estimators on three experimental datasets are shown in Table 4.

Table 4. Six-degree-of-free (6-DoF) pose estimation performance on three experimental datasets.

		BR	UWA3M	UWAOR
RANSAC	err_t	0.000	7.929	9.513
	err_r	0.030	0.696	0.769
LRF-Net	err_t	0.000	6.088	4.392
	err_r	0.024	0.608	0.405

Two salient observations can be made from the table. First, both RANSAC and our method manage to achieve accurate pose estimation results on the BR dataset that contains point cloud pairs with large overlapping ratios. However, our method only needs $\frac{1}{10}$ of the iterations required for RANSAC. Second, on more challenging datasets, i.e., UWA3M and UWAOR, our method significantly outperforms RANSAC. This demonstrates that LRF-Net can improve the accuracy and efficiency of RANSAC for 6-DoF pose estimation simultaneously.

4.3. Analysis Experiments

4.3.1. Verifying the Rationality of LRF-Net

To verify the rationality of the main technique components of our LRF-Net, we conduct the following experiments. As mentioned above, our LRF-Net contains two main parts: Estimating z -axis and x -axis. First, in order to verify the choice of normal vector for z -axis calculation, we replace the normal vector with the one regressing z -axis via a network shown in Figure 10 (dubbed "DR"). Second, to confirm the advantage of our x -axis technique, we perform analysis experiments from three aspects. (1) To prove the advantage of invariant geometric attributes, we replace the invariant geometric attributes with the combination of original points and z -axis (i.g., $[\mathbf{q}_i, \mathbf{z}(\mathbf{p})]$). Then, we calculate the x -axis in a weighted vector-sum manner. The former is dubbed "Sum1" and the latter is dubbed "Sum2". (2) In order to verify the choice of weighted vector-sum operation for x -axis calculation, we test the approach using the vector with the maximum weight as the x -axis (dubbed "Max"). (3) To demonstrate that the axes of LRF is not suitable to be directly regressed, we compare our method with the one regressing x -axis via a network (DR). There are totally eight different combinations. All of them are tested on BR, UWA3M and UWAOR datasets. The results are shown in Tables 5–7.

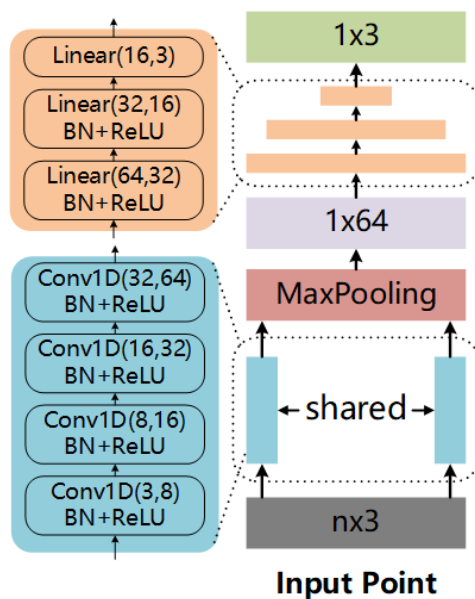


Figure 10. The architecture of DR.

Clearly, LRF-Net (Normal + Sum1) achieves the best performance among tested methods. It verifies that learning weights via invariant geometric attributes rather than directly learning axes is more reasonable. In addition, vector-sum is more appropriate for integrating projection vectors with learned weights for LRF-Net.

Table 5. MeanCos performance of eight different combinations on BR Dataset.

BR Dataset					
z-Axis	x-Axis	Sum1	Sum2	DR	Max
Normal		0.999	0.999	0.775	0.720
DR		0.737	0.778	0.582	0.471

The bold means the better performance compared with other methods.

Table 6. MeanCos performance of eight different combinations on UWA3M Dataset.

UWA3M Dataset					
z-Axis	x-Axis	Sum1	Sum2	DR	Max
Normal		0.690	0.429	0.574	0.412
DR		0.390	0.495	0.323	0.287

The bold means the better performance compared with other methods.

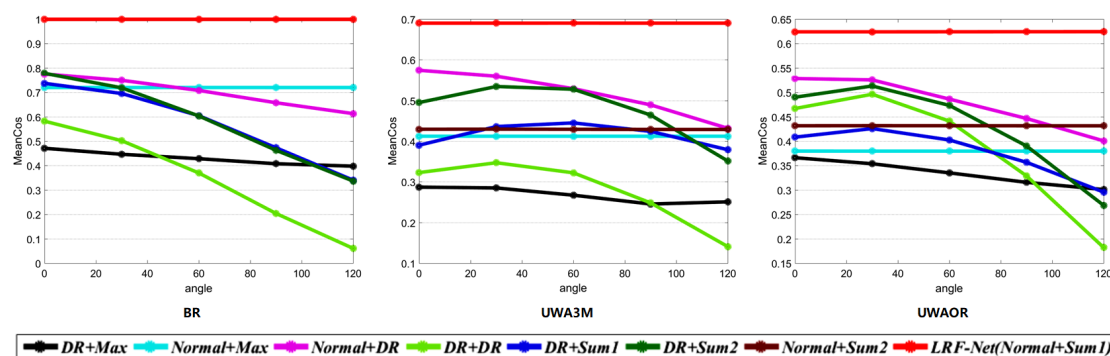
Table 7. MeanCos performance of eight different combinations on UWAOR Dataset.

UWAOR Dataset					
z-Axis	x-Axis	Sum1	Sum2	DR	Max
Normal		0.624	0.432	0.528	0.380
DR		0.408	0.490	0.467	0.366

The bold means the better performance compared with other methods.

4.3.2. Resistance to Rotation

To evaluate the robustness of LRF-Net to rotation, we manually rotate the tested data. Specifically, we rotate the scene point clouds a certain degree among z-axis (i.g., 30, 60, 90, and 120 degrees). Then, we measure their MeanCos performances. Figure 11 displays the results of eight different combinations.

**Figure 11.** Robustness performance of eight combination on three rotated datasets.

As shown in Figure 11, we can see that LRF-Net, Normal+Max, and Normal+Sum2 achieve very stable performances. The other ones which include "DR" part, show less robust performances.

This result has demonstrated two conclusions. One is that it is hard to achieve rotation-invariance by only relying on original points. A guidance (e.g., normal vector) is very necessary. The other is that the invariant attributes is not indispensable. Just a simple combination (e.g., combination of original

points and normal vector) can also achieve rotation-invariance. However, the invariant attributes can boost the performance of our network.

4.3.3. Performance under Varying Support Radius

Figure 12 shows MeanCos performances of six LRF methods under varying support radius on three public datasets without noise. From the observation of Figure 12, we can see that our LRFNet achieves a stable and outstanding performance on the BR dataset. On the UWA3M and UWAOR datasets, our LRFNet outperforms other LRF methods when support radius is more than 7.5 mr. Another observation is that the performance of our LRFNet is tending towards stability with the increase of support radius, while some other LRF methods present a downward trend. It verifies that our LRFNet is able to gain a stable LRF from a local surface which contains enough points to guarantee its statistical significance and uniqueness.

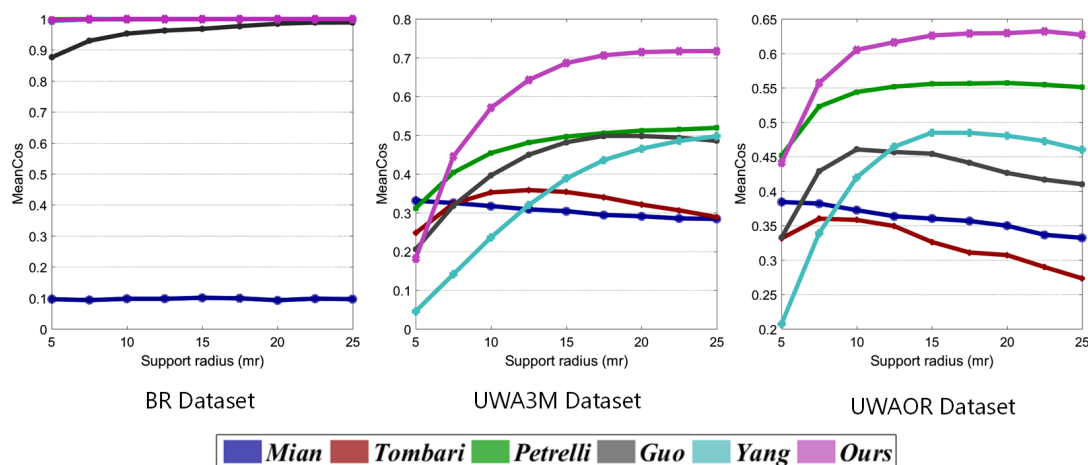


Figure 12. MeanCos performance of six LRF methods under varying support radius on three public dataset.

4.3.4. Visualization

Figure 13 visualizes the learned weights by our LRF-Net for several sample local surfaces, which presents two interesting findings. First, closer points do not seem to have greater contributions. It is a common assumption for many existing CA- and PSD-based LRF methods, including Tombari, Guo, and Yang, that closer points should have greater weights. However, they are inferior to our LRF-Net in terms of repeatability performance. Second, x -axis estimation is generally determined by a particular area, rather than a single salient point as employed by many PSD-based methods, e.g., Petrelli. These visualization results also demonstrate our opinion that each neighboring point in the local surface gives a unique contribution to LRF construction.

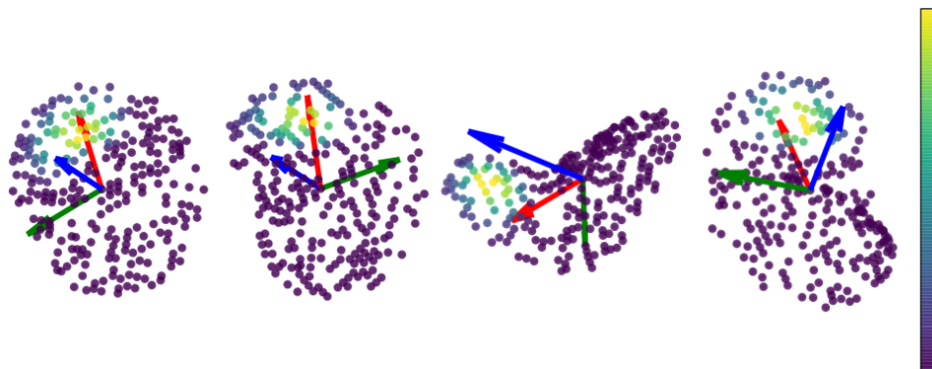


Figure 13. The visualization of the weights for every point in a local surface.

5. Conclusions

In this paper, we proposed LRF-Net, a learned LRF for 3D local surface that is repeatable and robust to a number of nuisances. LRF-Net assumes that each neighboring point in the local surface gives a unique contribution to LRF construction and measure such contributions via learned weights. Experiments showed that our LRF-Net outperforms many state-of-the-art LRF methods on datasets addressing different application scenarios. In addition, LRF-Net can significantly boost the local shape description and 6-DoF pose estimation performance.

6. Future Work

In the future, we are going to do our research in two interesting directions. The first one is to consider the texture of 3D objects. RGB information can provide a power guidance when the 3D models lack sufficient geometric features but has photometric cues. The other is to take multi-scale geometric information into consideration.

Author Contributions: A.Z. principal investigator, designed the project, data acquisition, performed analysis and wrote the manuscript. J.Y. supervised the project and revised the paper. Z.C. supervised the project and approved final submission. W.Z. contributed to data acquisition and diagnosis of the cases. All authors have read and agreed to the published version of the manuscript.

Funding: This work is jointly supported by the National Natural Science Foundation of China (Grant No. U1913602), the National Key R&D Program of China (No. 2018YFB1305504) and the Natural Science Basic Research Plan in Shaanxi Province of China (Grant No. 2020JQ-210).

Conflicts of Interest: The authors declare no conflict of interest.

References

- Guo, Y.; Sohel, F.; Bennamoun, M.; Lu, M.; Wan, J. Rotational projection statistics for 3D local surface description and object recognition. *Int. J. Comput. Vis.* **2013**, *105*, 63–86. [\[CrossRef\]](#)
- Petrelli, A.; Di Stefano, L. On the repeatability of the local reference frame for partial shape matching. In Proceedings of the IEEE International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2244–2251.
- Tombari, F.; Salti, S.; Di Stefano, L. Unique signatures of histograms for local surface description. In Proceedings of the European Conference on Computer Vision, Heraklion, Crete, Greece, 5–11 September 2011; Springer: Berlin/Heidelberg, Germany, 2010; pp. 356–369.
- Gojcic, Z.; Zhou, C.; Wegner, J.D.; Wieser, A. The perfect match: 3d point cloud matching with smoothed densities. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5545–5554.
- Spezialetti, R.; Salti, S.; Stefano, L.D. Learning an Effective Equivariant 3D Descriptor Without Supervision. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 6401–6410.
- Derpanis, K.G. Overview of the RANSAC Algorithm. *Image Rochester NY* **2010**, *4*, 2–3.
- Deng, H.; Birdal, T.; Ilic, S. 3D Local Features for Direct Pairwise Registration. *arXiv* **2019**, arXiv:1904.04281.
- Yang, J.; Xiao, Y.; Cao, Z. Toward the Repeatability and Robustness of the Local Reference Frame for 3D Shape Matching: An Evaluation. *IEEE Trans. Image Proces.* **2018**, *27*, 3766–3781. [\[CrossRef\]](#)
- Mian, A.; Bennamoun, M.; Owens, R. On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes. *Int. J. Comput. Vis.* **2010**, *89*, 348–361. [\[CrossRef\]](#)
- Petrelli, A.; Di Stefano, L. A repeatable and efficient canonical reference for surface matching. In Proceedings of the Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission, Zurich, Switzerland, 13–15 October 2012; pp. 403–410.
- Yang, J.; Zhang, Q.; Xiao, Y.; Cao, Z. TOLD: An effective and robust approach for 3D local shape description. *Pattern Recognit.* **2017**, *65*, 175–187. [\[CrossRef\]](#)
- Bro, R.; Acar, E.; Kolda, T.G. Resolving the sign ambiguity in the singular value decomposition. *J. Chemom.* **2008**, *22*, 135–140. [\[CrossRef\]](#)

13. Chua, C.S.; Jarvis, R. Point signatures: A new representation for 3d object recognition. *Int. J. Comput. Vis.* **1997**, *25*, 63–85. [[CrossRef](#)]
14. Deng, H.; Birdal, T.; Ilic, S. Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 602–618.
15. Tombari, F.; Salti, S.; Di Stefano, L. Performance evaluation of 3D keypoint detectors. *Int. J. Comput. Vis.* **2013**, *102*, 198–220. [[CrossRef](#)]
16. Mian, A.S.; Bennamoun, M.; Owens, R.A. A novel representation and feature matching algorithm for automatic pairwise registration of range images. *Int. J. Comput. Vis.* **2006**, *66*, 19–40. [[CrossRef](#)]
17. Mian, A.S.; Bennamoun, M.; Owens, R. Three-dimensional model-based object recognition and segmentation in cluttered scenes. *IEEE Trans. Pattern Anal. Mach. Intel.* **2006**, *28*, 1584–1601. [[CrossRef](#)] [[PubMed](#)]
18. Malassiotis, S.; Srinivasan, M.G. Snapshots: A novel local surface descriptor and matching algorithm for robust 3D surface alignment. *IEEE Trans. Pattern Anal. Mach. Intel.* **2007**, *29*, 1285–1290. [[CrossRef](#)] [[PubMed](#)]
19. Fischler, M.A.; Bolles, R.C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **1981**, *24*, 381–395. [[CrossRef](#)]
20. Guo, Y.; Bennamoun, M.; Soheli, F.; Lu, M.; Wan, J.; Kwok, N.M. A comprehensive performance evaluation of 3D local feature descriptors. *Int. J. Comput. Vis.* **2016**, *116*, 66–89. [[CrossRef](#)]
21. Rusu, R.B.; Cousins, S. 3d is here: Point cloud library (pcl). In Proceedings of the IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 1–4.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).