Information Sciences 520 (2020) 209–231

Contents lists available at ScienceDirect

Information Sciences

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

An efficient and compact 3D local descriptor based on the weighted height image

Tiecheng Sun, Guanghui Liu*, Shuaicheng Liu, Fanman Meng, Liaoyuan Zeng, Ru Li

School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, 611731, PR China

ARTICLE INFO

Article history:
Received 21 October 2019
Revised 31 January 2020
Accepted 5 February 2020
Available online 5 February 2020

Keywords:
3D local descriptor
Local reference frame
Point cloud registration
Weighted height image

ABSTRACT

3D local descriptors are the fundamental and essential elements that have been commonly applied in 3D computer vision. This paper proposes a novel and effective 3D local descriptor for describing the 3D local shape. The research focuses on accelerating the descriptor generation by simplifying the Local Reference Frame (LRF) and optimizing the feature space through a Weighted Height Image (WHI). An in-depth theoretical analysis of the LRF is conducted. Then, this study proposes a simplified LRF to reduce the redundant computations of the covariance matrix and share the calculations with the 3D information coding. Besides, the feature space is modeled and analyzed in this paper. Based on the analysis, we propose a weighting function to strengthen the abilities of the feature representation. The experimental results indicate that the proposed WHI descriptor outperforms the state-of-the-art (SOTA) algorithms in terms of accuracy and efficiency. Meanwhile, the compactness of the WHI is about six times more than that of the SOTA algorithms. Moreover, for the application of point cloud registration, the proposed WHI exhibits high effectiveness in terms of both accuracy and real-time capability.

1. Introduction

3D local descriptors are frequently used for describing the features of the local 3D shape in point cloud, which is a fundamental research topic in 3D vision. Similar to the features extracted from the 2D images that have been widely applied in image recognition [20,47] and image retrieval [19,46], 3D features have also been extensively used in 3D object recognition [4,13,15] and 3D object retrieval [1,14,35]. Moreover, they are the key technologies for point cloud registration [16,18,30] and 3D reconstruction [29,33,38,49]. However, obtaining an effective and compact 3D local descriptor is still a challenging task, which faces three major problems. First, the robustness and descriptiveness of the descriptors can be seriously affected by noise, occlusions and point cloud self-similarities, which frequently occurs in real applications. Second, the existing 3D local descriptors are time-consuming due to the large scale point clouds, which is not applicable for many real-time applications, such as automatic driving and robot navigation. Third, with the extensive applications of point cloud data, the processing, storage and transmission of mass data have become a new bottleneck. To this end, the scope of this paper focuses on the study of seeking a more robust, efficient, and compact 3D shape description method.

* Corresponding author.
E-mail addresses: tiechengsun@std.uestc.edu.cn (T. Sun), guanghuiliu@uestc.edu.cn (G. Liu), liushuaicheng@uestc.edu.cn (S. Liu), fmmeng@uestc.edu.cn (F. Meng), lyzeng@uestc.edu.cn (L. Zeng), liru@std.uestc.edu.cn (R. Li).

https://doi.org/10.1016/j.ins.2020.02.004
0020-0255/

3D shape description is a process of the 3D information coding. In the process, the first fundamental attribute that should be considered is the rotation invariance. Concerning this point, the existing 3D local descriptors can be classified into two categories, one is the Rotation-Invariant Metrics (RIM) based methods and the other is the Local Reference Frame (LRF) based methods. In general, the RIM based methods directly encode the particularly designed rotation-invariant metrics, such as angles, distances, circular bins and frequency domain coding. The classical algorithms include the Point Feature Histograms (PFH) [31], Fast Point Feature Histograms (FPFH) [30], Spin Image (SI) [21], Log-Polar Height Map (LPHM) [25], Local Feature Statistics Histogram (LFSH) [40], to name a few. However, these RIM based methods suffer from the following problems. Firstly, some metrics are not robust. For example, angles and frequency domain coding are sensitive to noise. Secondly, the calculation of some metrics is complex and time-consuming, such as the PFH and FPFH. Thirdly, most of the coding is irreversible, i.e., the metrics calculation through information coding would result in a loss of the original information.

In contrast, the LRF based methods have obvious advantages. On one hand, these methods estimate a rotation-invariant local coordinate system, i.e., LRF, which is more repeatable and robust to occlusions and clutter. On the other hand, benefitting from the LRF, the 3D information coding process becomes easier due to the absence of the rotation invariance consideration, which allows the coding process to retain the original information. The classical LRF based algorithms include the Unique Shape Context (USC) [35,36], Signature of Histograms of OrienTations (SHOT) [33,36], Rotational Projection Statistics (RoPS) [15,17], Triple Orthogonal Local Depth Images (TOLDI) [42], to name a few. Due to the advantages of the LRF, the LRF based methods have become a research hotspot in recent years. However, the estimation of the LRF increases the time cost of the descriptor. For most of the existing methods, the LRF estimation and the 3D information coding are two independent processes, within which redundant computations are involved, limiting the efficiency. For the sake of efficiency, some methods adopted 2D forms for 3D representation. Because 2D forms have fewer dimensions and thus, more efficient than 3D forms, such as the snapshots [24], LPHM [25] and TOLDI [42]. However, these methods ignore the special properties of the feature space, which sacrifices the abilities of the feature representation.

Based on the above investigations, it is worth noting that the LRF estimation is time-consuming, and there has been little research focuses on the feature space modeling and optimization. Therefore, the motivation of this paper is to enhance the LRF based methods by simplifying the calculations as well as optimizing the feature space. A more robust, efficient and compact 3D shape descriptor is proposed. Firstly, with respect to efficiency, a simplified LRF is proposed by reducing the redundant calculations. Moreover, special modifications are designed such that the LRF estimation and the 3D information coding can share some calculations. Regarding the feature space optimization, a weighting function is proposed through mathematical modeling and analysis, such that the abilities of the feature representation can be strengthened.

In particular, firstly, we make a theoretical analysis of the EigenValue Decomposition (EVD) based LRF, which shows that a robust LRF estimation is an ill-conditioned problem. This is because the axes are determined by the difference of the eigenvalues. This difference is not an absolute but a relative difference, i.e., the ratio between the eigenvalues. Therefore, omitting the normalization factor of the covariance matrix will not change the ratio between the eigenvalues. Furthermore, in the case of occlusions, the covariance matrix calculated in a small region is more robust than in a larger one. While in the case of no occlusion, the smaller region will not decrease the robustness of the LRF. So we can properly reduce the region corresponding to the covariance matrix. Therefore, in this paper, we simplify the LRF from the above two aspects for improving the overall efficiency of the descriptor. At the same time, since the signs of eigenvectors obtained by EVD are ambiguous [3], we propose a sign disambiguation method by summing the different coordinates of the neighboring points, which makes some calculations in the LRF be reused in the 3D information coding.

Secondly, a Weighted Height Image (WHI) method is proposed to make the information coding function be close to well-condition, and then further reduce the negative influence of the coordinate change caused by the LRF estimation error. Compared with the state-of-the-art (SOTA) algorithms, a comprehensive evaluation of performance is conducted on three popular datasets (Bologna, Kinect and Space time [36]). Experimental results show that the proposed method outperforms the SOTA methods.

Finally, a WHI based point cloud registration method is proposed following the coarse-to-fine strategy [40,45], achieving high accuracy and real-time performance, which further proves the effectiveness of the proposed method. The major contributions of this paper are summarized as follows:

(1) A simplified LRF is proposed, which is fast and robust. Meanwhile, some important calculations of the LRF can be reused in the 3D information coding. As such, the construction of the proposed descriptor becomes more efficient.
(2) A robust, efficient and compact descriptor based on the WHI is presented, which is close to well-condition in feature space.
(3) A coarse-to-fine point cloud registration method based on the proposed descriptor is proposed, which is real-time and precise.

Based on the preliminary version, Local Height Image Descriptor (LHID) [34], we make further theoretical analysis and essential algorithm improvement. The remainder of the paper is organized as follows. Section 2 gives a brief review of the related popular algorithms. Section 3 theoretically analyzes the LRF and 2D representation of 3D shape, then introduces the proposed method in detail. Section 4 introduces the evaluation criteria, datasets and experimental setup, then records the experimental results and analysis. Section 5 introduces the application of WHI on point cloud registration. Section 6 concludes this paper and discusses future work.

2. Related work

2.1. Rotation-invariant metrics based methods

In the past decades, many descriptors based on the rotation-invariant metrics have been proposed. Such algorithms encode the 3D information while calculating the metrics. These metrics mainly include angles, distances, circular bins and frequency domain coding. For example, based on the metric of angles, researchers proposed the THRIFT [10], FPFH [30], LFSH [40] and Statistic of Deviation Angles on Subdivided Space (SDASS) [48]. Based on circular bins, the SI [21] and LFSH were proposed. While the LPHM [25] and 3D Shape Context (3DSC) [11] were presented based on the rotation invariance of the frequency domain coding. It is worth noting that in most of the above methods, the calculation of the metrics is usually based on the normal vector or local reference axis. However, these metrics also have some drawbacks. For example, the calculation of the complex metrics is time-consuming, such as the PFH and FPFH. Meanwhile, some metrics such as the angles and frequency domain coding are sensitive to noise. Also, most of the metrics are irreversible, meaning that the original 3D information cannot be pushed back from the calculated metrics. For example, the histogram composed of angles cannot inversely deduce the original local point cloud geometry. So, there is a loss of the information in the metrics calculation.

2.2. Local reference frame based methods

Generally, the LRF is estimated based on the eigenvalue decomposition of the covariance matrix. However, the signs of the eigenvectors obtained by EVD are ambiguous [3]. In order to get a unique LRF, Tombari et al. [36] proposed reorientation of the eigenvectors according to the consistency of the vector directions from the neighboring points to the basis point. Guo et al. [17] presented a scatter matrix based LRF, which is robust to varying mesh resolution, occlusions and clutter. Yang et al. [42] proposed first to solve the Z-axis with EVD, and then calculate the projection vectors and integrate them to get X-axis. The cross-product between Z-axis and X-axis obtains the Y-axis. There are also some other modifications of the LRF, refer to [29,41,50].

The LRF has three uniquely determined axes, which makes it rotation-invariant. Therefore, the descriptors based on the LRF have gained wide attention. For example, Frome et al. [11] established a spherical coordinate system based on the normal. Then, they divided the sphere into bins and accumulated a weighted count to generate the 3DSC descriptor. Due to the lack of the LRF, the 3DSC needs to calculate multiple descriptions for every feature point. Then, Tombari et al. [35,36] proposed the USC to improve the 3DSC based on the LRF. Compared with the 3DSC, the USC achieves significant improvement on accuracy and RAM usage. Another LRF based descriptor SHOT [33,36] was also proposed to map angles histograms into an isotropic spherical grid for seeking a trade-off between descriptiveness and robustness. Some other LRF based methods such as the RoPS [15,17] and TOLDI [42] have also achieved superior performance. These methods usually include two processes, i.e., the LRF estimation and 3D information coding. However, in the existing methods, these two steps are independent, which limits the efficiency of the descriptors. Furthermore, because the LRF is obtained by EVD and the signs need to be redefined, the existing LRF is still time-consuming.

2.3. 3D information coding

Information coding is the re-expression of the original data. The coding process has a clear objective in different applications as introduced in [19,44]. The 3D information coding is another important factor that should be considered in generating descriptors with respect to the LRF. We divide the coding methods into two categories, i.e., the signature and histogram [33]. The signature is defined as mapping the trait values to coordinates, while the histogram is defined as the count of the trait values. Spatial division based histograms are commonly used for the descriptors. For example, Johnson and Hebert [21] proposed to divide the cylinder space into rings by radius and elevation. The histogram is generated by a 2D accumulator. As for the 3DSC [11] and USC [35], the bins are obtained in spherical coordinates. However, these methods of space division will produce invalid bins without any points. Larger bins can reduce the number of empty ones, but this will result in a lower resolution for a descriptor. Another type of histogram is obtained by the statistics of the metrics. For example, the PFH [31] and its variation FPFH [30] use angles to construct histograms. However, the calculation of the metrics usually results in a loss of geometric information. Some metrics such as the angles are sensitive to noise. While the signature based methods usually map the 3D information to a coordinate system. For example, the SHOT [33,36] maps the histograms into an isotropic spherical grid, producing a hybrid descriptor between the signatures and histograms. Similarly, Guo et al. [17] rotated the point cloud by several angles and projected the point cloud onto the three planes determined by the LRF, then combined all the statistics of the distribution matrixes into the descriptors. So this is also a hybrid descriptor. Another typical signature based method is to use 2D images to encode the 3D information, which is more compact. The compact feature coding methods have high efficiency [7,24,25,39,43]. For example, Yang et al. [39] proposed to use the georeferenced feature image to represent a 3D LiDAR point cloud. As for the descriptors, Malassiotis and Strintzis [24] proposed to project the 3D points onto the virtual image plane. Masuda [25] proposed to map the height values onto the tangent plane and transform the coordinate system to log-polar for further encoding. While Yang et al. [42] projected the 3D points on the three orthometric planes determined by the LRF for the information coding. Unlike RIM based methods, these de-

scriptors preserve more of the original information. However, the existing methods usually ignore the special properties of the feature space, which limits the expression capability of the feature space.

In this paper, we combine the LRF and the 2D image representation to improve the accuracy, efficiency and compactness of the descriptor. Different from the methods in [24,25,42], we propose a more robust and efficient simplified LRF, which makes the LRF estimation and the 3D information coding share the calculations. Moreover, a weighting function is also proposed to improve the capabilities of the feature representation.

3. Analysis and methodology

The flowchart of the proposed descriptor is shown in Fig. 1. It consists of two procedures, i.e., the LRF estimation and the 3D information coding. In Fig. 1, the first procedure includes steps 1–5 as shown in the orange dashed line. The second procedure includes steps 1 and 4–10 as shown in the green dashed line. The detailed flow of the algorithm as shown in Fig. 1 is introduced as the following steps:

0. The inputs include the feature point and the corresponding neighboring points.
1. The translation-invariant vectors and the distances are calculated from the feature point to its neighboring points.
2. The above two metrics are used for constructing a simplified covariance matrix.
3. After EVD of the covariance matrix, three axes are obtained with ambiguous signs.
4. All the translation-invariant vectors are projected onto *X* and *Y* axes.
5. The sign of each axis is determined by the sum of the projected values with respect to each direction.
6. After signs disambiguation, the unique LRF and the projected points without ambiguous signs are obtained.
7. A weighting function is calculated for optimizing the feature.
8. The projected points are used for generating a height image. Then the WHI is obtained by using the weighting function to weight the height image.
9. The generated WHI is smoothed by a Gaussian filter.
10. The output is the vector expanded by the WHI.

In steps 1, 4 and 5, as shown in Fig. 1, the calculations are shared between the LRF estimation and the 3D information coding, which makes the algorithm more efficient.

In this section, we first give a detailed theoretical analysis of the LRF estimation, then propose the simplified strategy. Moreover, a new sign disambiguation method by summing the different coordinates is presented. For the 3D information coding, the coded feature space is modeled and analyzed in detail. Finally, a weighting function is presented for optimizing the feature space.

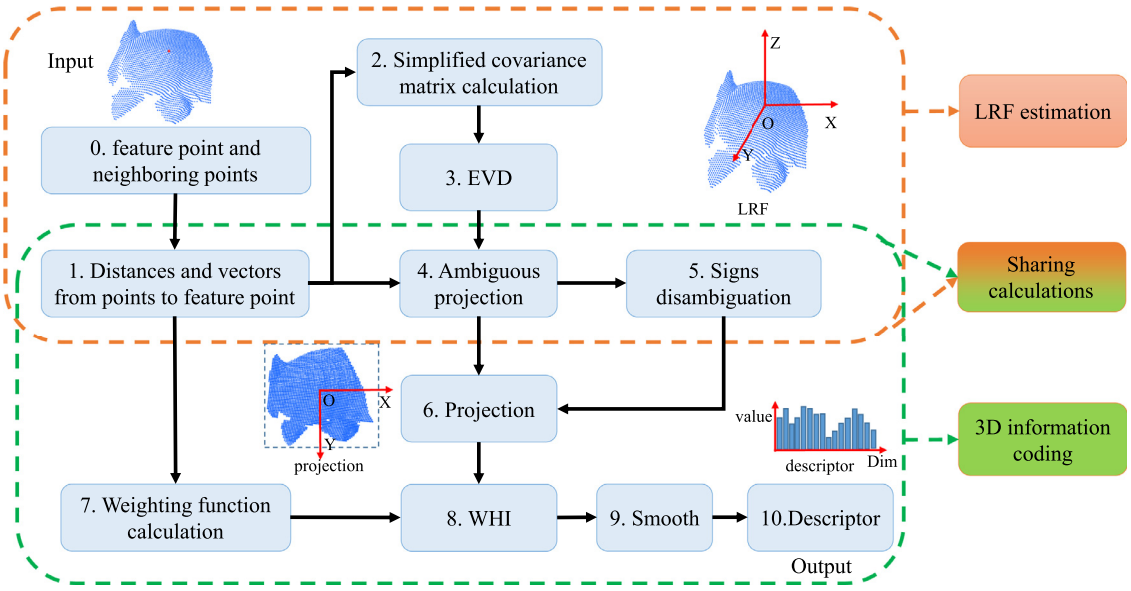

Fig. 1. Flowchart of the proposed method. The algorithm consists of two procedures, the LRF estimation and 3D information coding. These two procedures share the calculations of steps 1, 4 and 5, which are illustrated as the intersection of the two different colored dashed lines. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.1. LRF Estimation

3.1.1. Covariance matrix analysis

LRF is both translational and rotational invariant. Thus, it is widely used in many popular descriptors such as the SHOT [36], USC [35], RoPS [17] and TOLDI [42]. For the clearness, we first analyze the translation and rotation invariance. In three-dimensional space $\mathbb{R}^3$, $\mathbf{p} = [x\ y\ z]$ denotes a point. Hence, in another three-dimensional space $\mathbb{R}'^3$, the coordinate of this point can be expressed as:

$$\mathbf{p}' = [x\ y\ z][\mathbf{v}_x\ \mathbf{v}_y\ \mathbf{v}_z] + \mathbf{t} = \mathbf{pR} + \mathbf{t}, \tag{1}$$

where $\mathbf{v}_x$, $\mathbf{v}_y$ and $\mathbf{v}_z$ are the column vectors of matrix $\mathbf{R}$. Furthermore, $\mathbf{R}$ and $\mathbf{t}$ are the rotation matrix and translation vector, respectively. Eq. (1) means that $\mathbb{R}^3$ is transformed to $\mathbb{R}'^3$ by $\mathbf{R}$ and $\mathbf{t}$. For the translation invariance, the most common and effective metric is the difference between the two points, denoted by $\mathbf{p}_{ij} = \mathbf{p}_i - \mathbf{p}_j$. The point after translation is defined as $\mathbf{p}' = \mathbf{p} + \mathbf{t}$. Then, the translation invariance can be represented by:

$$\mathbf{p}'_{ij} = \mathbf{p}'_i - \mathbf{p}'_j = (\mathbf{p}_i + \mathbf{t}) - (\mathbf{p}_j + \mathbf{t}) = \mathbf{p}_i - \mathbf{p}_j = \mathbf{p}_{ij}. \tag{2}$$

After the LRF being determined on the feature point $\mathbf{p}_f$, the neighboring points $\mathbf{p}_i$ around $\mathbf{p}_f$ can be expressed as $\mathbf{p}_{if}$. Therefore, the translation invariance of the LRF based methods is guaranteed by the translation-invariant vectors $\mathbf{p}_{if}$.

For rotation invariance, the new reference frame named LRF can be simply interpreted as the inverse of the rotation matrix $\mathbf{R}$, denoted by $\mathbf{R}^{-1}$. We define the point after rotation as $\mathbf{p}' = \mathbf{pR}$. Then, the rotation invariance can be expressed as:

$$\mathbf{p}'' = \mathbf{p}'\mathbf{R}^{-1} = \mathbf{pRR}^{-1} = \mathbf{p}. \tag{3}$$

The LRF is generally obtained by EVD of the covariance matrix [26]. The covariance matrix is expressed as:

$$\mathbf{C}(\widehat{\mathbf{p}}) = \frac{1}{k}\sum_{i=0}^{k}(\mathbf{p}_i - \widehat{\mathbf{p}})^{\mathrm{T}}(\mathbf{p}_i - \widehat{\mathbf{p}}), \tag{4}$$

where k is the number of points in the support region, $\hat{\mathbf{p}}$ represents the barycenter of the neighboring points around the feature point $\mathbf{p}_f$ in the support region. To increase efficiency, Tombari et al. [36] substitute the feature point $\mathbf{p}_f$ for $\hat{\mathbf{p}}$. At the same time, they calculate a weighted covariance matrix to resist clutter and occlusions, as:

$$\mathbf{C}(\mathbf{p}_f) = \frac{1}{\sum_{i:d_i\le R}(R - d_i)}\sum_{i:d_i\le R}(R - d_i)\big(\mathbf{p}_i - \mathbf{p}_f\big)^{\mathrm{T}}\big(\mathbf{p}_i - \mathbf{p}_f\big), \tag{5}$$

where R represents the support radius and $d_i = \|\mathbf{p}_i - \mathbf{p}_f\|_2$. Replacing $\hat{\mathbf{p}}$ with $\mathbf{p}_f$ not only improves efficiency but also improves the robustness. In order to demonstrate this point, we assume that an accurate LRF of a feature point $\mathbf{p}_f$ can be obtained by giving N neighboring points in the support region. If we move one point $\mathbf{p}_m$ slightly in the support region, the barycenter will also change slightly. Here, $\hat{\mathbf{p}}'$ denotes the changed barycenter. According to Eq. (2), the following formulas hold:

$$\hat{\mathbf{p}}' \neq \hat{\mathbf{p}} \Rightarrow \mathbf{p}'_i - \hat{\mathbf{p}}' \neq \mathbf{p}_i - \hat{\mathbf{p}}, \quad \forall i, \tag{6}$$

$$\begin{cases} \mathbf{p}'_f = \mathbf{p}_f \quad \mathbf{p}'_i = \mathbf{p}_i \Rightarrow \mathbf{p}'_i - \mathbf{p}'_f = \mathbf{p}_i - \mathbf{p}_f, & i \neq m \\ \mathbf{p}'_f = \mathbf{p}_f \quad \mathbf{p}'_i \neq \mathbf{p}_i \Rightarrow \mathbf{p}'_i - \mathbf{p}'_f \neq \mathbf{p}_i - \mathbf{p}_f, & i = m. \end{cases} \tag{7}$$

Eq. (6) shows that all the terms of the sum in Eq. (4) have been changed slightly, while Eq. (7) shows that there is only one that has been changed in Eq. (5), which indicates that the estimation error of $\mathbf{C}(\hat{\mathbf{p}})$ is larger than that of $\mathbf{C}(\mathbf{p}_f)$. In fact, $\mathbf{p}_i$, $\mathbf{p}_f$ and $\hat{\mathbf{p}}$ can be considered as independent random variables. We assume that $\mathbf{p}_i$, $\mathbf{p}_f$ and $\hat{\mathbf{p}}$ all follow Gaussian distributions. Random variables $\mathbf{p}_i$ and $\mathbf{p}_f$ have the same distribution with zero mean and the variance of σ_1^2. However, $\hat{\mathbf{p}}$ is more sensitive to occlusions as illustrated in Fig. 2, so the variance of $\hat{\mathbf{p}}$, σ_2^2, is larger than σ_1^2, i.e., $\sigma_2^2 > \sigma_1^2$. The variance of the random variables $(\mathbf{p}_i - \hat{\mathbf{p}})$ and $(\mathbf{p}_i - \mathbf{p}_f)$ can then be expressed as:

$$D\big(\mathbf{p}_i - \hat{\mathbf{p}}\big) = \sigma_1^2 + \sigma_2^2, \tag{8}$$

$$D\big(\mathbf{p}_i - \mathbf{p}_f\big) = \sigma_1^2 + \sigma_1^2 = 2\sigma_1^2. \tag{9}$$

Obviously, $D(\mathbf{p}_i - \hat{\mathbf{p}}) > D(\mathbf{p}_i - \mathbf{p}_f)$, indicating that in the case of occlusion, the error of the covariance matrix obtained by Eq. (4) is larger than that of Eq. (5). Therefore, $\mathbf{p}_f$ is more suitable for generating the vectors and calculating the covariance matrix.

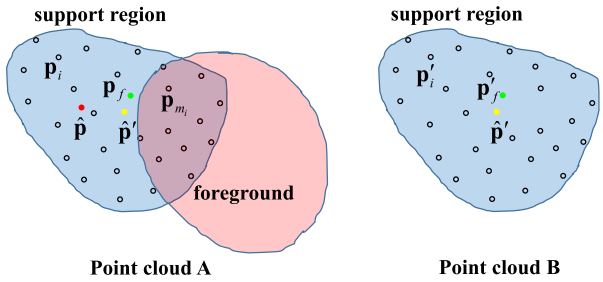

Fig. 2. When point cloud A has occlusions, the overlap between the support region and the foreground does not exist. In this case, the barycenter of the neighboring points in the support region changes significantly. The notations $\hat{\mathbf{p}}$ and $\hat{\mathbf{p}}'$ represent the barycenter of the point cloud A and B, respectively. In contrast, the feature point $\mathbf{p}_f$ remains stationary regardless of the occlusions.

3.1.2. Simplified LRF

In this section, we first present the simplified LRF, and then exemplify the modifications in detail based on the analysis. On the basis of the covariance matrix in Eq. (5) [36], we propose a further simplification, expressed as:

$$\mathbf{C}(\mathbf{p}_f) = \sum_{i:d_i \le \mu R} (R - d_i)(\mathbf{p}_i - \mathbf{p}_f)^{\mathrm{T}}(\mathbf{p}_i - \mathbf{p}_f), \tag{10}$$

where $0 < \mu \le 1$. The factor μ determines the region corresponding to $\mathbf{C}(\mathbf{p}_f)$. The major modifications are reducing the region of the covariance matrix and omitting the normalization factor.

Firstly, the analysis in Section 3.1.1 shows that when there exists occlusions, the $\mathbf{p}_f$ based covariance matrix $\mathbf{C}(\mathbf{p}_f)$ in Eq. (5) is more robust than $\hat{\mathbf{p}}$ based covariance matrix $\mathbf{C}(\hat{\mathbf{p}})$ in Eq. (4). Therefore, we adopt the method based on the feature point $\mathbf{p}_f$. However, the occlusions cause part of the points to disappear as illustrated in Fig. 2. So, Eq. (7) becomes

$$\begin{cases} \mathbf{p}'_f = \mathbf{p}_f \quad \mathbf{p}'_i = \mathbf{p}_i \Rightarrow \mathbf{p}'_i - \mathbf{p}'_f = \mathbf{p}_i - \mathbf{p}_f, & i \neq m_1, m_2, \cdots, m_n \\ \mathbf{p}'_f = \mathbf{p}_f \quad \mathbf{p}'_i \neq \mathbf{p}_i \Rightarrow \mathbf{p}'_i - \mathbf{p}'_f \neq \mathbf{p}_i - \mathbf{p}_f = \mathbf{0}, & i = m_1, m_2, \cdots, m_n. \end{cases} \tag{11}$$

In Fig. 2, since the points $\mathbf{p}_{m_i}$ are occluded in the point cloud A, the vectors $\mathbf{p}_{m_i f} = \mathbf{p}_{m_i} - \mathbf{p}_f$ are equivalent to zero vectors, which increase the error of the covariance matrix calculation. Therefore, $\mathbf{p}_{m_i}$ can be considered as noise. In order to solve this problem, we reduce the region corresponding to the covariance matrix by a factor of μ (set to 0.7 in this paper). As shown in Fig. 3, the reduction greatly reduces the interference caused by occlusions. This further accelerates the calculation of the covariance matrix by reducing the number of points. When there is no occlusion, the points reduction will not affect the robustness of the LRF.

Secondly, we further simplify the LRF by omitting the normalization factor $\frac{1}{\sum_{i:d_i \le R}(R-d_i)}$ in Eq. (5). Since $\mathbf{C}(\mathbf{p}_f)$ is a real symmetric matrix, and its EVD can be expressed as:

$$\mathbf{C}(\mathbf{p}_f) = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T, \tag{12}$$

where $\mathbf{Q} = [\boldsymbol{\gamma}_1\ \boldsymbol{\gamma}_2\ \boldsymbol{\gamma}_3]$ is an orthogonal matrix, $\boldsymbol{\gamma}_1$, $\boldsymbol{\gamma}_2$ and $\boldsymbol{\gamma}_3$ are the unit orthogonal eigenvectors. The notation $\mathbf{\Lambda}$ is a diagonal matrix composed of the eigenvalues λ_1, λ_2, and λ_3 ($\lambda_1 > \lambda_2 > \lambda_3$). As described in [42], the local geometry is more likely to be flat or symmetric. The *X*-axis of the LRF is harder to estimate than the *Z*-axis. This is because the eigenvalues represent variances in spatial dimensions, and λ_3 is usually smaller than λ_1 and λ_2, so the *Z*-axis is easily distinguished

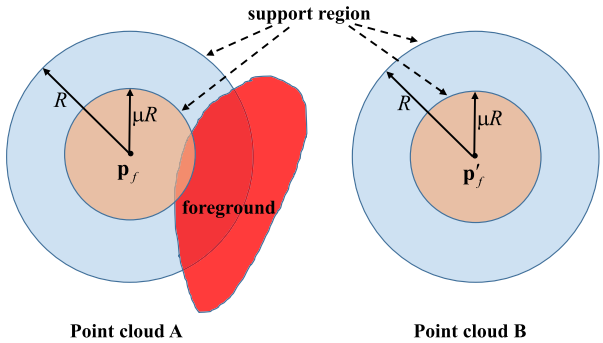

Fig. 3. Point cloud A has occlusions. The figure shows that if we reduce the support region, the proportion of the interference (the overlap between the support region and the foreground) caused by occlusions will decrease or even disappear.

from the other two axes. Since λ_1 and λ_2 have similar values, the X-axis is difficult to estimate accurately. The axes are determined by the difference of the eigenvalues. When we remove the normalization factor, the scale of the eigenvalues will change. Nevertheless, the proportion between the eigenvalues does not change, as well as eigenvectors. This means that it is the ratio between the eigenvalues determines the axes. We define the condition number as:

$$\begin{cases} cond_{xy} = \frac{\lambda_1}{\lambda_2} \\ cond_{xz} = \frac{\lambda_1}{\lambda_3}. \end{cases} \tag{13}$$

A larger condition number means that the two directions are easier to distinguish and the LRF estimation is more accurate. Therefore, the LRF estimation is an ill-conditioned problem. Eq. (13) indicates that the condition number is not a function of the normalization factor, so we can omit it to accelerate the calculation.

3.1.3. Disambiguation of sign

After EVD of the covariance matrix, the signs of eigenvectors (i.e., $\boldsymbol{\gamma}_1$, $\boldsymbol{\gamma}_2$ and $\boldsymbol{\gamma}_3$) are ambiguous according to Bro et al. [3]. In order to obtain a unique LRF, the signs should be redetermined. We use the geometric properties to reorient the eigenvectors. As discussed in Section 3.1.1, the points in the LRF are represented by the translation-invariant vectors $\mathbf{p}_{if}$ for the translation invariance. For the rotation invariance, the LRF can be expressed as an inverse of the rotation matrix. Therefore, we define it as $\mathbf{R}_{\text{am}}^{-1} = [\boldsymbol{\gamma}_x\ \boldsymbol{\gamma}_y\ \boldsymbol{\gamma}_z]$ with ambiguous signs, where $\boldsymbol{\gamma}_x = \boldsymbol{\gamma}_1$, $\boldsymbol{\gamma}_z = \boldsymbol{\gamma}_3$ and $\boldsymbol{\gamma}_y$ is the cross product between $\boldsymbol{\gamma}_1$ and $\boldsymbol{\gamma}_3$. Then, the coordinates of the neighboring points can be expressed as:

$$\mathbf{p}_{if}\mathbf{R}_{\text{am}}^{-1} = \mathbf{p}_{if}\left[\boldsymbol{\gamma}_x\ \boldsymbol{\gamma}_y\ \boldsymbol{\gamma}_z\right] = \left[\mathbf{p}_{if}\cdot\boldsymbol{\gamma}_x\ \mathbf{p}_{if}\cdot\boldsymbol{\gamma}_y\ \mathbf{p}_{if}\cdot\boldsymbol{\gamma}_z\right] = \left[x'_{i\_\text{am}}\ y'_{i\_\text{am}}\ z'_{i\_\text{am}}\right]. \tag{14}$$

In order to disambiguate the signs, different coordinates are summed as $sum_{x'} = \sum_{i:d_i<R} x'_{i\_\text{am}}$ and $sum_{z'} = \sum_{i:d_i<R} z'_{i\_\text{am}}$. Then, we can get the sign of each vector in $\mathbf{R}_{\text{am}}^{-1}$ as follows:

$$sign_x = \begin{cases} 1 & sum_{x'} >= 0 \\ -1 & sum_{x'} < 0 \end{cases}, \tag{15}$$

$$sign_z = \begin{cases} 1 & sum_{z'} >= 0 \\ -1 & sum_{z'} < 0 \end{cases}, \tag{16}$$

$$sign_y = sign_x \times sign_z. \tag{17}$$

Finally, we define a sign vector as $\mathbf{s} = [sign_x\ sign_y\ sign_z]^{\mathrm{T}}$. Then the final coordinates of the points in the support region can be expressed as

$$\mathbf{p}'_{if} = \mathbf{p}_{if}\mathbf{R}^{-1} = \left[sign_x \times x'_{i\_\text{am}}\ sign_y \times y'_{i\_\text{am}}\ sign_z \times z'_{i\_\text{am}}\right] = \left[x'_i\ y'_i\ z'_i\right]. \tag{18}$$

Supposing that $\mathbf{R}^{-1}$ is the disambiguation result of $\mathbf{R}_{\text{am}}^{-1}$, the unique LRF can be denoted by $\mathbf{R}^{-1} = \mathbf{R}_{\text{am}}^{-1}\mathbf{s}$.

3.2. WHI generation

3.2.1. Coding function analysis

The 3D information coding based on 2D forms has been applied in many descriptors, such as the snapshots [24], LPHM [25], TOLDI [42] and LHID [34]. According to the coding method, we define a coding function (denoted by $\mathscr{F}$) for each descriptor. In general, the 3D local shape can be considered as a surface in 3D space and expressed as $\mathscr{F}(x, y, z) = 0$. According to Malassiotis and Strintzis [24], the coding function of the snapshots can be expressed as:

$$\mathscr{F}_z(x, y) = z. \tag{19}$$

Similar to the snapshots, the methods LPHM [25] and LHID [34] also use the same coding function. Nevertheless, the TOLDI [42] uses the combination of $\mathscr{F}_z(x, y) = z$, $\mathscr{F}_x(y, z) = x$ and $\mathscr{F}_y(z, x) = y$ as the coding function. These coding functions take the feature point $\mathbf{p}_f$ as the origin. In the LRF, if the coordinates of a point are $[x_i, y_i, z_i]$, the corresponding value of the function is:

$$\mathcal{F}_z(x_i, y_i) = z_i = \mathbf{p}_{if}\cdot\boldsymbol{\gamma}_z = \left(\mathbf{p}_i - \mathbf{p}_f\right)\cdot\boldsymbol{\gamma}_z. \tag{20}$$

If the point is $\mathbf{p}_f$, then,

$$\mathcal{F}_z\left(x_f, y_f\right) = \left(\mathbf{p}_f - \mathbf{p}_f\right)\cdot\boldsymbol{\gamma}_z = 0. \tag{21}$$

We assume that the local surface is continuous, then the function $\mathscr{F}_z(x, y) = z$ is also continuous, which can be expressed by $\varepsilon - \delta$, i.e., for any small value ε, there always exists a δ, yielding

$$\left|\mathscr{F}_z(x, y) - \mathscr{F}_z\left(x_f, y_f\right)\right| < \varepsilon, \tag{22}$$

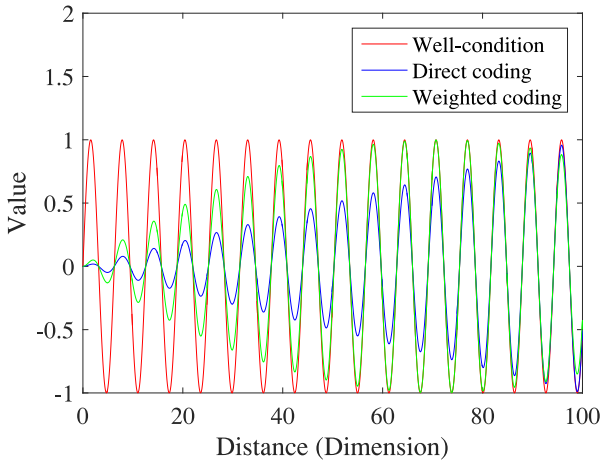

Fig. 4. The three curves illustrate three different conditions of the feature space. The red curve is the ideal well-condition. It means that every dimension has the same variance. The blue curve indicates the condition of directly mapping the 3D information to the corresponding 2D feature space. The green curve represents the condition of the feature space optimized by the weighted coding function. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

when $\sqrt{(x-x_f)^2+(y-y_f)^2} < \delta$. This indicates that the value of the coding function $\mathscr{F}_z(x,y)$ is always close to zero near $\mathbf{p}_f$. As (x, y) moves away from the origin, the amplitude of the function $\mathscr{F}_z(x,y)$ increases gradually.

For simplicity, the mathematical modeling and equivalent analysis of the coding function are performed in 2D space. Then, the equivalent coding function is denoted by $\mathscr{F}(x)$, $x \in [0, 1]$. Supposing that the coded feature space $\mho^n$ has n dimensions, the ith dimension represents the function feature corresponding to $x = i/n$. We also assume that the feature space is well-conditioned. It means that if each dimension is treated as a random variable, their variances are equal. As illustrated in Fig. 4, the red curve represents the well-condition. The sequence $(\omega_1, \omega_2, \cdots, \omega_n)$ denotes the feature vectors of the well-conditioned feature space. However, the feature space has special property as shown in Eq. (22), which is abstracted and simplified to the blue curve in Fig. 4. Nevertheless, in order to simplify the analysis, we assume that the blue curve is obtained by multiplying the envelope function $f(x) = x$ by the well-conditioned feature. As shown by the blue curve in Fig. 4, the fluctuation amplitude of each dimension changes due to the continuity of the coding function. We define this property as a system, of which the response is a new feature space, expressed as:

$$\mathbf{f} = \sum_{i=1}^{n} \omega_i r_i \mathbf{e}_i = \begin{pmatrix} r_1 & 0 & \cdots & 0 \\ 0 & r_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & r_n \end{pmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_n \end{bmatrix} = \mathbf{A}\omega, \tag{23}$$

where $\mathbf{e}_i$ is the direction vector of $\mho^n$ and the finite sequence $(r_1, r_2, \cdots, r_n)$ defines this system. In this system, we set $r_i = i/n$. Ultimately, the system is expressed as matrix $\mathbf{A}$, the input is ω and the output is the system response $\mathbf{f}$ that is illustrated by the blue curve in Fig. 4. If we set $r_i = 1$, the corresponding system response is the red curve in Fig. 4. According to Eq. (23), the system can be equivalent to a diagonal transformation matrix $\mathbf{A}$ composed of the sequence $(r_1, r_2, \cdots, r_n)$. The red and blue curves in Fig. 4 indicate that matrix $\mathbf{A}$ determines the variance of each dimension of the feature space. Therefore, the sequence $(r_1, r_2, \cdots, r_n)$ can be considered as the eigenvalues of the system.

In the feature space, the distance metrics are used for the matching of the features. We give two commonly used distance metrics

$$d_1(\mathbf{f}, \mathbf{f}') = \left|\mathbf{f} - \mathbf{f}'\right| = \sum_{i=1}^{n} \left|\omega_i - \omega'_i\right| r_i \|\mathbf{e}_i\| = \sum_{i}^{n} \varepsilon_i r_i, \tag{24}$$

$$d_2(\mathbf{f}, \mathbf{f}') = \left\|\mathbf{f} - \mathbf{f}'\right\|_2 = \sqrt{\sum_{i=1}^{n} (r_i(\omega_i - \omega'_i))^2} = \sqrt{\sum_{i=1}^{n} (r_i \varepsilon_i)^2}, \tag{25}$$

where $\varepsilon_i = (\omega_i - \omega'_i)$ is the difference of the ith dimension between two features. For $d_1(\mathbf{f}, \mathbf{f}')$ or $d_2(\mathbf{f}, \mathbf{f}')$, ε_1 always corresponds to a smaller weight r_1, however, ε_n corresponds to a larger one r_n. Assuming that $\varepsilon_1 = \varepsilon_n$, the change of $d_1(\mathbf{f}, \mathbf{f}')$ or $d_2(\mathbf{f}, \mathbf{f}')$ caused by ε_1 will be much smaller than that caused by ε_n. This indicates that the current feature space is not well-conditioned.

In particular, we define the condition number to describe each dimension. The sequence $(r_1, r_2, \cdots, r_n)$ are the eigenvalues of the diagonal matrix $\mathbf{A}$, so $\lambda_i = r_i$. The notation $\lambda_{\max}$ denotes the largest eigenvalue. Then, the condition number of

each dimension can be defined as:

$$cond_i = \frac{\lambda_{\max}}{\lambda_i}. \tag{26}$$

When each dimension has a similar effect on the distance metrics, the feature space is more expressive, and the condition number $cond_i$ is closer to one. Therefore, this problem is a well-conditioned problem.

3.2.2. Weighted coding function

In this paper, in order to make the feature space close to well-condition, we propose a weighting function, which is defined as:

$$\mathscr{W}(x, y, z) = \eta + (1 - \eta)\frac{(R - d_i(x, y, z))}{R}, \quad 0 < \eta \leq 1, \tag{27}$$

where η is a factor that controls the range, R is the support radius, and $d_i(x, y, z)$ is the distance between $\mathbf{p}_f$ and $\mathbf{p}_i$. Then, we optimize the coding function with this weighting function. The weighted coding function is defined as:

$$\mathscr{F}(x, y, z) = \mathscr{W}(x, y, z)\mathscr{F}_z(x, y) = \left[\left(\eta + (1 - \eta)\frac{(R - d_i)}{R}\right) \times z_{(x,y)}\right]. \tag{28}$$

In order to illustrate the role of the weighted coding function, we take the 2D space described in Section 3.2.1 as the premise for analysis. In Eq. (23), the sequence $(r_1, r_2, \cdots, r_n)$ that forms the system matrix $\mathbf{A}$ can be considered as the samples of function $f(x) = x$. Similarly, as for the weighted coding function Eq. (28), the corresponding sequence can also be considered as the samples of function $f'(x)$. In order to simplify the analysis, we set $R = 1$, $\eta = 0.3$. In the equivalent 2D space, d_i is equivalent to x. Then, the equivalent coding function becomes

$$f'(x) = (1 - 0.7x) \times f(x) = (1 - 0.7x)x. \tag{29}$$

We assume that when condition $c_{\text{well}} : cond_i < \frac{4}{3}$ is true, the corresponding ith dimension is well-conditioned. Fig. 5 illustrates the differences between the two coding functions. As shown in Fig. 5, the value of the function represents the eigenvalue, so the maximum value of the function is the maximum eigenvalue $\lambda_{\max}$. Fig. 5(a) shows that for the function $f(x) = x$, the region that satisfies the condition c_{well} is

$$\Delta_{X1} = X_B - X_A = \frac{1}{4}, \tag{30}$$

while for $f'(x)$, Fig. 5(b) shows that the region is $\Delta_{X2} = \frac{9}{14}$. Obviously, $\Delta_{X2} > \Delta_{X1}$. This means that the function $f'(x)$ has a larger region that satisfies the condition c_{well}. In other words, compared with $f(x)$, the weighted coding function makes more dimensions close to well-condition in the feature space. In order to further compare the two coding functions, we normalize the weighted coding function and draw the system response in Fig. 4, shown in green. In Fig. 4, the weighted coding function (the green curve) is closer to the well-condition (the red curve) than the original one (the blue curve).

The weighting function $\mathscr{W}$ can not only optimize the condition numbers but also resist the estimation error of X-Y coordinate system in the LRF. As analyzed in Section 3.1.2, λ_1 is close to λ_2. Therefore, the estimated X and Y axes are likely to deflect at an angle as shown in Fig. 6. In the coordinate system in Fig. 6, the error of a point coordinate is proportional

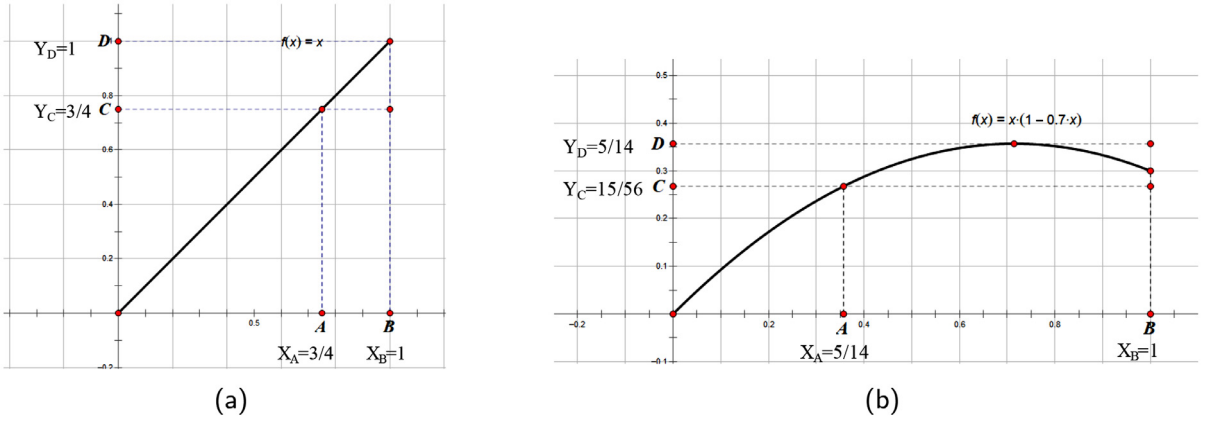

Fig. 5. Condition number analysis: In the feature space, the condition number of each dimension is defined as the ratio of the maximum eigenvalue to the corresponding eigenvalue, expressed as $cond_i = \frac{\lambda_{\max}}{\lambda_i}$. The condition number can be expressed as $cond_i = \frac{\lambda_{\max}}{f(x)}$, where the maximum eigenvalue is the maximum value of the function. We assume that when $cond_i < 4/3$, the feature is well-conditioned. For function (a), the region corresponding to the well-condition is $X_B - X_A = 1/4$. For function (b), $X_B - X_A = 9/14$.

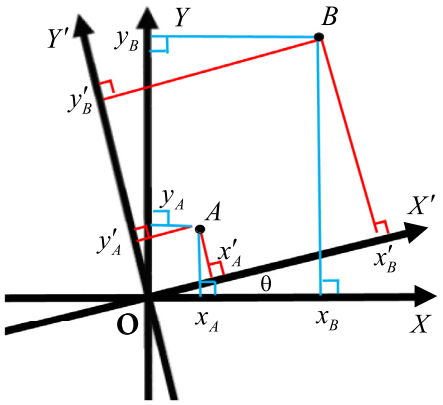

Fig. 6. When the coordinate system is rotated by a certain angle, the coordinates of the points will change. The points that are near the origin change less than those that are far away. For example, in the *XOY* coordinate system, the coordinates of the points *A* and *B* are (x_A, y_A) and (x_B, y_B), respectively. While in *X′OY′*, the coordinates are (x'_A, y'_A) and (x'_B, y'_B), respectively. Obviously, $|x_B - x'_B| > |x_A - x'_A|$ and $|y_B - y'_B| > |y_A - y'_A|$.

to the distance from the point to the origin. Therefore, the coordinates error can be reduced if the points far from the origin have smaller weights. Fortunately, the weighting function $\mathscr{W}$ can suppress this error simultaneously.

Remark. We use an approximate equivalence analogy to analyze the problem. For example, we use $f(x) = x$ to approximate the equivalent system corresponding to the code function $\mathscr{F}_z(x, y) = z$. To an actual surface with random distribution, the above analysis shows that the proposed weighting function can improve the expression of the original surface. The optimization theory is satisfied with the general function $f(x)$ in most cases.

3.2.3. Generation of the WHI descriptor

In order to generate descriptors, the coding function should be encoded again or directly discretized into descriptors. For example, the snapshots [24] and TOLDI [42] directly map the coding function to descriptors. While the LPHM [25] encodes the coding function $\mathscr{F}_z(x, y) = z$ twice. The descriptor is discretized by the lattice. In the existing methods [24,25,42], the value of the lattice is usually defined as the shortest projection distance to resist the occlusions. When the neighborhood of a local surface is much smaller than the whole object, the function $\mathscr{F}_z(x, y) = z$ is generally single-valued. However, the functions $\mathscr{F}_x(y, z) = x$ and $\mathscr{F}_y(z, x) = y$ are usually multi-valued. So in this paper, the pixel (lattice) value of the image is determined by the mean of all the weighted heights rather than the minimum distance. Then, the proposed descriptor is defined as a Weighted Height Image (WHI),

$$WHI(x, y) = \frac{1}{k(x, y)} \sum_{(\mathbf{p}_{if}\cdot\boldsymbol{\gamma}_x, \mathbf{p}_{if}\cdot\boldsymbol{\gamma}_y)\in WHI(x,y)} \mathcal{W}(x, y, z) \times \left(\mathbf{p}_{if} \cdot \boldsymbol{\gamma}_z\right), \tag{31}$$

where $k(x, y)$ is the number of the points projected onto the lattice $WHI(x, y)$. We let $z' = \mathbf{p}_{if} \cdot \boldsymbol{\gamma}_z$, which is a random variable. Since the point clouds are easily disturbed by noise, we assume that the mean of the random noise n is zero. Then the expectation of random variable $(z' + n)$ can be expressed as:

$$E\left[z' + n\right] = E\left[z'\right] + E[n] = E\left[z'\right]. \tag{32}$$

This indicates that the statistical property of mean is robust to the zero-mean noise.

To increase the efficiency of the WHI, we directly reuse the calculations x', y' and z' obtained in Eq. (18). Moreover, the distances $d_i(x, y, z)$ calculated in Eq. (10) are shared with the weighting function in Eq. (27). Therefore, only a few weights and mean calculations are required in the generation of the WHI. Attributed to the sharing calculations between the LRF and the 3D information coding, our method is very efficient.

In the discretization process, some pixels may have no corresponding projected points. For the purpose of filling these "holes", the method snapshots [24] uses linear interpolation, while the TOLDI [42] fills the "holes" with a larger value. Since the continuity of $\mathscr{F}(x, y)$ may be broken by the discretization or noise, we use the Gaussian filter to smooth the WHI for restoring the continuity. We assume that the random surface is uniformly distributed in *X-Y* plane. The probability density function is

$$f(x, y) = \begin{cases} \frac{1}{4R^2}, & (x, y) \in D \\ 0, & (x, y) \notin D \end{cases}, \tag{33}$$

where D is the rectangular region of the WHI, $D : -R \le x \le R, -R \le y \le R$. The edge probability density of the random variable $\mathbf{X}$ is

$$f_{\mathbf{X}}(x) = \int_{-R}^{R} \frac{1}{4R^2} dy = \frac{1}{2R}. \tag{34}$$

Similarly, $f_{\mathbf{Y}}(y) = \frac{1}{2R}$. Therefore, $\mathbf{X}$ and $\mathbf{Y}$ are both uniformly distributed. So their means $\mathrm{E}(\mathbf{X}) = \mathrm{E}(\mathbf{Y}) = 0$, and the variances $\sigma_x^2 = \sigma_y^2 = \frac{4R^2}{12} = \frac{R^2}{3}$. Then, we define the Gaussian filter as:

$$H(u,v) = \exp\left(-\frac{u^2}{2\sigma_x^2} - \frac{v^2}{2\sigma_y^2}\right) = \exp\left(-\frac{3\left(u^2+v^2\right)}{2R^2}\right). \tag{35}$$

Hence, we can obtain the kernel coefficients with different kernel sizes, which is an optional parameter. The effect of this parameter on the performance of the descriptor will be analyzed in detail in Section 4.3.

Compared with similar methods such as the snapshots [24], LPHM [25] and TOLDI [42], our method has at least four advantages, summarized as follows:

(1) Our WHI is equipped with our presented simplified LRF.
(2) For coding function, the snapshots [24] directly uses the function $\mathscr{F}_z(x,y) = z$. The TOLDI [42] combines multi-view information for coding. For rotation-invariance, the LPHM [25] carries out log-polar coordinate transformation and Fourier expansion. The coding function $\mathscr{F}_z(x,y) = z$ is encoded again. Our method uses a weighted coding function, which is close to well-condition and can reduce the adverse impact caused by the estimation error of X and Y axes in the LRF.
(3) In the process of discretization, the pixel value is defined as the distance of the nearest point in the snapshots [24], LPHM [25] and TOLDI [42]. While in our method, the pixel value is defined as the mean value, which is robust to the zero-mean noise.
(4) In order to fill the "holes", the snapshots [24] uses linear interpolation. While the TOLDI [42] fills them with a large value. Considering the continuity of the coding function, we smooth the WHI with the Gaussian filter.

4. Experiments

In this section, we study the accuracy, efficiency and compactness of the descriptors. We first introduce the evaluation criteria, the experimental setup and the datasets. Then, the performance of the proposed descriptor is evaluated by comparing with several SOTA algorithms.

4.1. Evaluation criteria

Precision-Recall Curve (PRC) [27] is one of the most popular criteria used for evaluating the descriptiveness of the descriptors. As described in [14], PRC is generated by using the Nearest Neighbor Distance Ratio (NNDR) technique [23,27]. Specifically, the features of the source and target point cloud are first extracted, respectively. According to NNDR, for each feature $\mathbf{f}_i^{\mathrm{S}}$ of the point $\mathbf{p}_i^{\mathrm{S}}$ in the source, we find the nearest and second nearest neighbors (denoted by $\mathbf{f}_{i1}^{\mathrm{T}}$ and $\mathbf{f}_{i2}^{\mathrm{T}}$, corresponding to $\mathbf{p}_{i1}^{\mathrm{T}}$ and $\mathbf{p}_{i2}^{\mathrm{T}}$, respectively) in the target. Then, the distance ratio is defined as

$$r = \frac{\| \mathbf{f}_i^{\mathrm{S}} - \mathbf{f}_{i1}^{\mathrm{T}} \|_2}{\| \mathbf{f}_i^{\mathrm{S}} - \mathbf{f}_{i2}^{\mathrm{T}} \|_2}, \tag{36}$$

where the operation $\| \cdot \|_2$ represents the Euclidean distance in the feature space. When r is less than a certain threshold τ, the satisfying points are selected as candidate points. The number of them is denoted by N_{match}. If the distances between the candidate points and the ground-truth are smaller than another threshold τ_g, such matches are considered as correct. The number of the correct matches is denoted by N_{correct}. The total number of the points that are used for matching is defined as N_{total}. Then, we can calculate the *precision* rate and *recall* rate, expressed as:

$$Precision = \frac{N_{\mathrm{correct}}}{N_{\mathrm{match}}}, \tag{37}$$

$$Recall = \frac{N_{\mathrm{correct}}}{N_{\mathrm{total}}}. \tag{38}$$

Finally, RPC is drawn by changing the threshold τ from 0 to 1.0.

The area under PRC denoted by AUC_{pr} [6] is another popular criterion to evaluate the robustness [14] and the compactness [14,50] of the descriptors. In this paper, we use AUC_{pr} to represent the accuracy of the descriptors.

4.2. Datasets and experimental setup

We use the public descriptor matching datasets provided in [36][1] to validate the algorithms, including Bologna, Kinect and Space time datasets as shown in Figs. 7 and 8. All the datasets contain several scenes and models, where the scenes

[1] http://www.vision.deis.unibo.it/research/78-cvlab/80-shot.

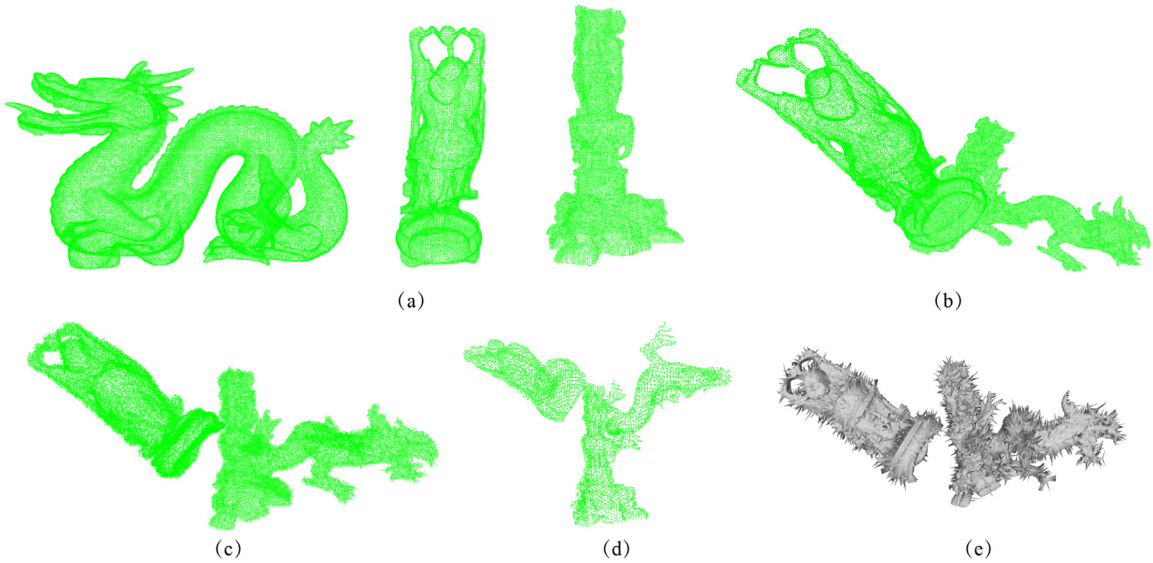

Fig. 7. Balogna dataset and its modifications. The dataset consists of models and scenes. The scene is composed of several models. The figures show (a) models, (b) scene, (c) Gaussian noise data, (d) decimation data and (e) shot noises data.

Fig. 8. Two 2.5D datasets, (a) Space time dataset, (b) Kinect dataset. Every dataset consists of models and scenes. The scene is generated by several models with background.

are composed of a series of models. We modify the Bologna dataset to evaluate the performance of the robustness and the descriptiveness on the Gaussian noise, varying mesh resolution and shot noises according to Guo et al. [14]. For a comprehensive evaluation, we also calculate AUC_{pr} performance on the popular 2.5D datasets Space time and Kinect [37]. In the matching process, instead of detecting the feature points, we randomly sample 1000 points in a model and extract the corresponding points in the scene as the feature points. Then, the features of the scene and the models are extracted and matched. We use the Fast Library for Approximate Nearest Neighbors (FLANN) [28] to realize fast feature matching. For comparison, based on Point Cloud Library (PCL) [32] (version 1.8.1), we compare the performance with several SOTA methods such as the SI [21], 3DSC [11], USC [35], SHOT [36] RoPS [15], LFSH [40] and TOLDI [42]. We run our experiment on a 3.4 GHz Intel(R) Core(TM) i5-7500 processor with 8 GB RAM.

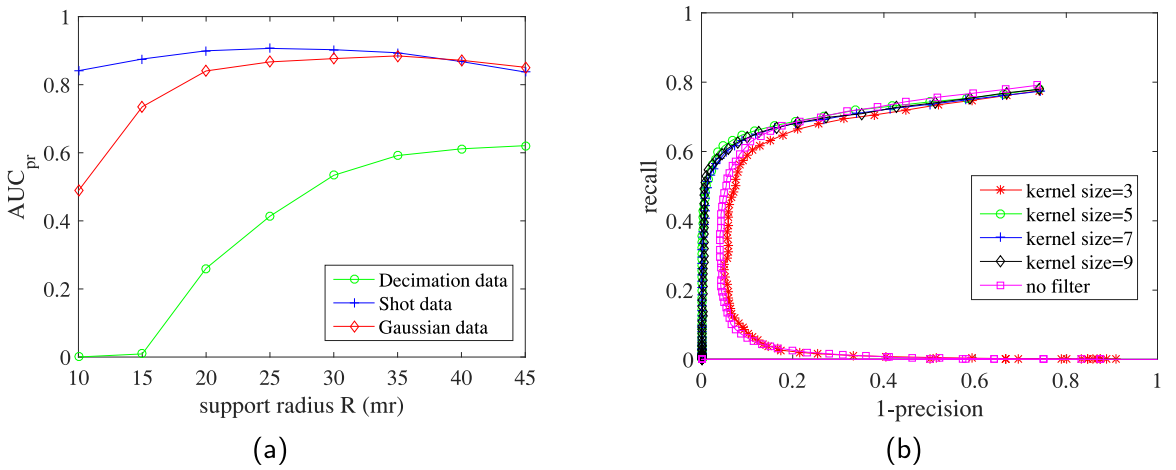

Fig. 9. WHI parameters analysis: (a) AUC_{pr} performance with different support radii. (b) PRC performance with different kernel size.

4.3. Parameters analysis

4.3.1. Support radius

The main purpose of this experiment is to analyze the accuracy with different support radii. So that we can set an appropriate radius R to make the descriptors perform well. We use AUC_{pr} to evaluate the accuracy, as shown in Fig. 9(a). In this experiment, we set the size of the WHI to $w_s \times w_s = 12 \times 12 = 144$. All radii are measured in mesh resolution (mr) of the original point cloud. The test decimation data is obtained by downsampling the scene to 1/32 of the original mesh resolution.

The results shown in Fig. 9(a) indicate that the accuracy of the WHI improves as the support radius increases on the Gaussian and decimation datasets, while for the shot noises data, AUC_{pr} is not sensitive to the support radius. When R approaches 20 mr, AUC_{pr} is close to the highest for the shot data. For the Gaussian data, the corresponding optimal radius is approximately 25 mr. While for the decimation data, the optimal radius is approximately 45 mr, which is larger than that of the shot noise and the Gaussian noise data. This is because if the support radius is not increased, there will be fewer points in the support region due to the low resolution of the decimation data. Therefore, in order to make the decimation data have higher accuracy, the support radius should be larger.

As stated in [42], a relatively large R performs well because there is more information available to discriminate the features. Thus, we also analyze the accuracy of several SOTA descriptors with different support radii as shown in Fig. 10. For each support radius, we add and average the values of AUC_{pr} of all the descriptors. Then, we find that for the Gaussian noise data, when the support radius is 30 mr, all the descriptors can perform well. For the shot noises and the decimation data, the appropriate radii are 25 mr and 45 mr, respectively.

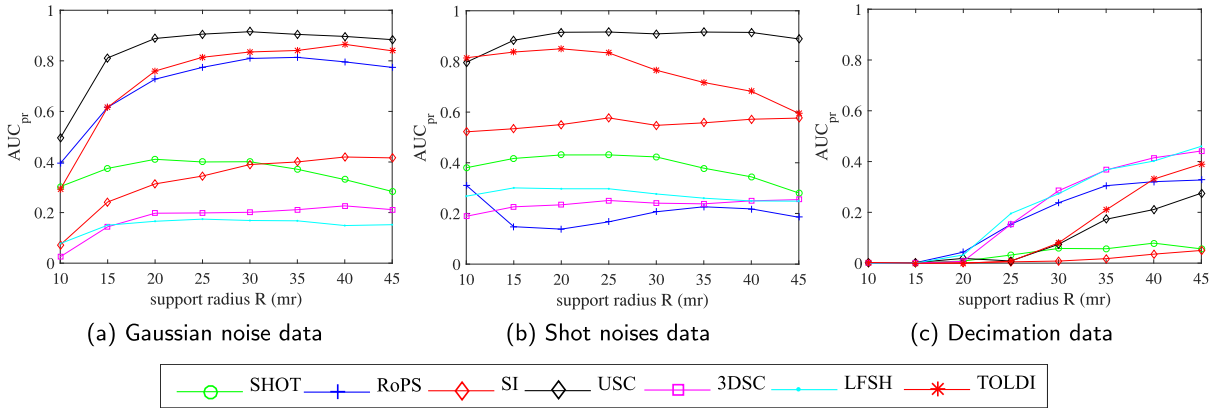

Fig. 10. AUC_{pr} performance of descriptors with different support radii on the (a) Gaussian noise, (b) shot noises and (c) decimation data.

4.3.2. The kernel size

As introduced in Section 3.2.3, the kernel size of the Gaussian filter is an optional parameter. The descriptiveness is analyzed with different kernel sizes by PRC, as shown in Fig. 9(b). The curves show that for the kernel size of 3 or without a smooth filter, the precision rate does not increase but decreases at a lower recall rate. When the kernel size is larger than 5, the descriptiveness increases. It means that the Gaussian filter can improve the descriptiveness of the WHI. Considering the complexity of the filtering, we prefer to choose a smaller kernel size, so we set the kernel size to 5×5.

4.4. Accuracy

In this section, we first use the RPC [27] to evaluate the descriptiveness performance. Then, we use multiple experimental RPCs to illustrate the robustness. Finally, we calculate AUC_{pr} for overall accuracy evaluation. For the sake of fairness, we ensure that every step in the evaluation the same, including the common parameters such as the support radius. The support radius is measured in mr. In order to make a comprehensive evaluation of the WHI with different dimensions, the size of the WHI is set to 4×4, 6×6, 8×8, 10×10, 12×12, 16×16 and 20×20, denoted by WHI16, WHI36, WHI64, WHI100, WHI144, WHI256 and WHI400, respectively.

4.4.1. Gaussian noise data

As illustrated in Section 4.3.1, a larger radius yields higher PRC performance. For the purpose of objectively evaluating the accuracy of the descriptors, the support radius should be appropriate for all the methods. So in our experiment, the support radius R is set to 30 mr according to the experimental results reported in Fig. 10(a). We add the Gaussian noise to the Bologna dataset with the standard deviation of 0.1 mr, 0.3 mr, 1.0 mr and 2.0 mr. Fig. 7(c) is an example of Gaussian noise data. Fig. 11 shows the RPC results. When the noise is smaller, such as 0.1 mr, 0.3 mr, the USC, RoPS, TOLDI and ours have similar descriptiveness performance. When the standard deviation increases to 1.0 mr or 2.0 mr, our descriptors outperforms the others. However, the TOLDI and RoPS descriptors are sensitive to the larger Gaussian noises. These figures show that the USC and our method are robust to the Gaussian noise. Moreover, the WHI16 also achieves better descriptiveness and robustness, which suggests that our lower dimensional WHI is still effective. If the point cloud is disturbed with the Gaussian noise, the angles between the points will change greatly. Therefore, the angle-based LFSH descriptor is sensitive to the Gaussian noise as shown in Fig. 11.

4.4.2. Shot noises data

As shown in Fig. 7(e), the shot noises points [14] are the outliers with a certain amplitude in the point cloud. We generate the outliers in a ratio of 2.0% and 5.0% with 3 mr and 6 mr amplitude in the scene, respectively. In this experiment, we set the support radius R to 25 mr according to the experimental results reported in Fig. 10(b). Fig. 12 illustrates that for the PRC performance, the USC is superior to the others, while the TOLDI and ours is close to the USC. Specially, the PRC performance of our WHI400 is very close to that of the USC in the four experiments. We find that in our proposed descriptors, the performance improves as the dimension increases. The LFSH is also sensitive to the shot noises. According to the four curves, with the increase of the outliers and the noise amplitude, the performance of the RoPS decreases substantially. The four figures show that the USC, TOLDI and WHI are robust to the shot noises.

4.4.3. Cross resolutions data

In order to evaluate the performance on different resolutions, we re-sample the scene to 1/4, 1/8, 1/16 and 1/32 of the original density as shown in Fig. 7(d). The models remain at the original resolution. As aforementioned in Section 4.3.1, cross resolutions experiment requires a larger support radius. This is because the premise of extracting a feature is the existence of the neighboring points in the support region. Therefore, the support radius R must be larger than the density of the decimation data. We define the original resolution of the models as 1 mr. So, the support radius is set to 35 mr for the first three cross resolutions experiments and 45 mr for the last one. Fig. 13 shows the PRC results. As for 1/4 and 1/8 decimation, the USC is superior to the others. The RoPS, TOLDI and our descriptors have a similar performance following the USC. When the rate of the decimation reaches 1/16 or 1/32, our descriptors outperform the others, followed by the RoPS and TOLDI. The four PRC curves in Fig. 13 illustrate that our WHI is more robust than the others. While the SHOT, SI, LFSH and 3DSC are sensitive to the cross resolutions data.

4.4.4. Comprehensive evaluation

In order to evaluate the performance more comprehensively, various experiments have been conducted through varying components of methods as well as test datasets. First, the Gaussian noise and varying mesh resolution are combined to create more challenging testing scenarios. Second, we test two other challenging datasets, Space time and Kinect dataset [37]. Fig. 8(a) shows the Space time dataset, its surface is smooth and difficult to discriminate. Fig. 8(b) shows the Kinect dataset acquired by Microsoft Kinect v1. The data is of low quality with noise. Both of these datasets are 2.5D with serious occlusions. Third, in order to analyze the performance of the LRF, we test the proposed WHI equipped with different LRFs proposed in [36] and [42], named as LRF(SHOT)+WHI and LRF(TOLDI)+WHI, respectively. Then, to verify the effectiveness of the proposed weighting function, we remove the weighting function in WHI400, named as HI400. Moreover, we weight the corresponding view of the TOLDI by a similar weighting function for further evaluation.

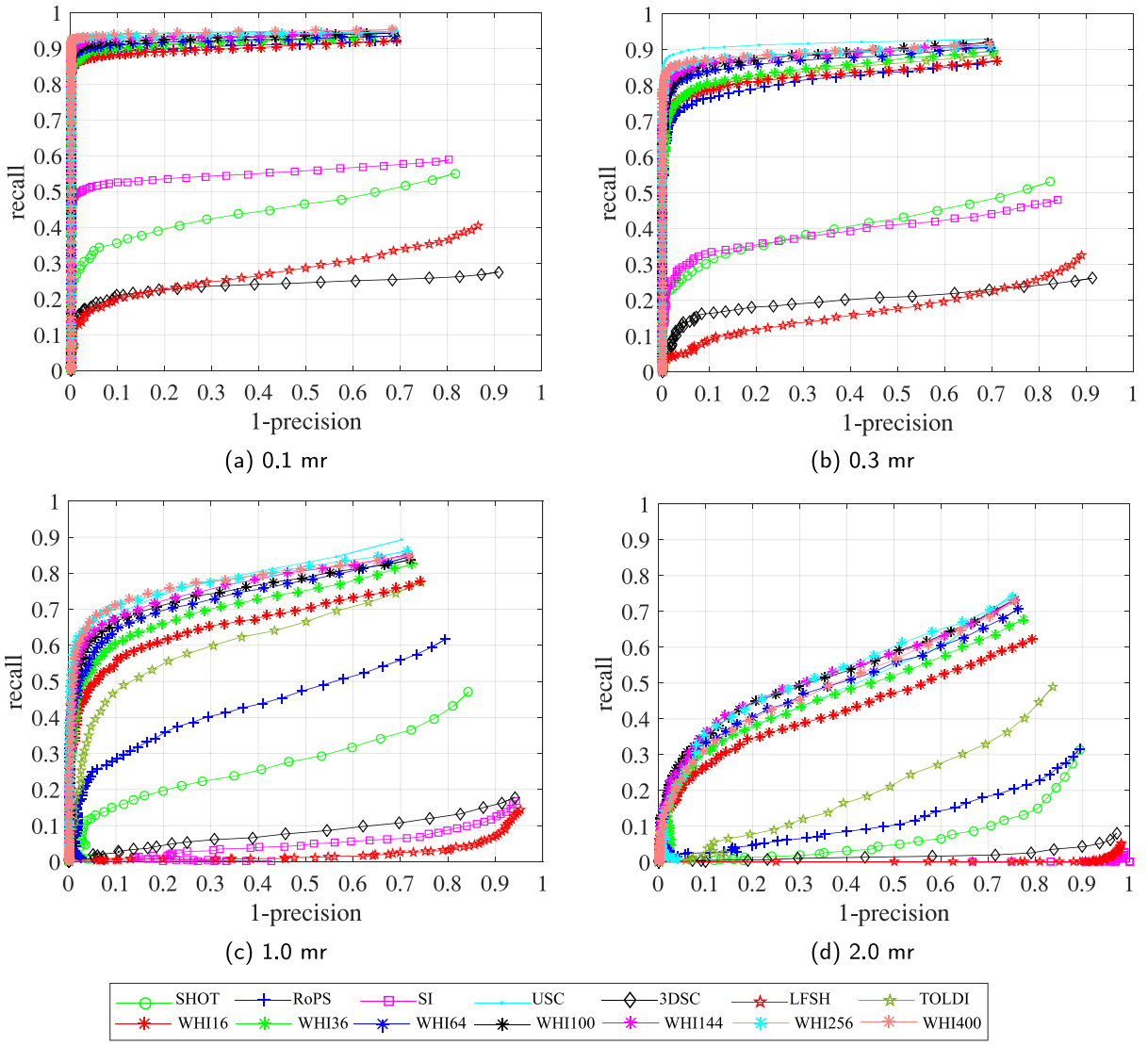

Fig. 11. Gaussian noise experiment: Gaussian noises with standard deviation of (a) 0.1 mr, (b) 0.3 mr, (c) 1.0 mr and (d) 2.0 mr are added to the Bologna data, respectively. The figures show the PRC curves.

Table 1 shows the AUC_{pr} performance of the above challenging datasets and the dimension of each descriptor. The results show that our method outperforms the other competitors on the Gaussian, decimation, Gaussian+decimation, Kinect and Space time datasets. While for the shot noises data, the USC achieves the best performance due to its high-dimensional space partition that makes the shot noises only interfere with a small number of dimensions by sacrificing the compactness. Nevertheless, our method adopts smooth filtering, which is not only robust to the shot noises but also compact. In terms of AUC_{pr} performance, our WHI is very close to that of the USC and superior to the others.

In order to verify the effectiveness of the proposed simplified LRF, we compare WHI400, LRF(SHOT)+WHI400 and LRF(TOLDI)+WHI400 in Table 1. The comparison between WHI400 and LRF(SHOT)+WHI400 indicates that the performance of the proposed simplified LRF is very close to that of SHOT [36] for the Gaussian, decimation and shot noises datasets. For the occluded datasets, such as Kinect and Space time, our simplified strategy is more robust than the SHOT as analyzed in Section 3.1.2. With regard to the LRF(TOLDI)+WHI400, the AUC_{pr} performance is close to the TOLDI and lower than WHI400, which indicates that the performance is limited by the LRF proposed in [42].

For the weighting function, the accuracy of WHI400 is higher than that of HI400 by a large margin, which shows that the proposed weighting function plays an obvious role in improving the feature space expression capabilities. The comparisons of the AUC_{pr} performance between the TOLDI, W+TOLDI and LRF(TOLDI)+WHI400 further illustrate the effectiveness of the proposed weighting function.

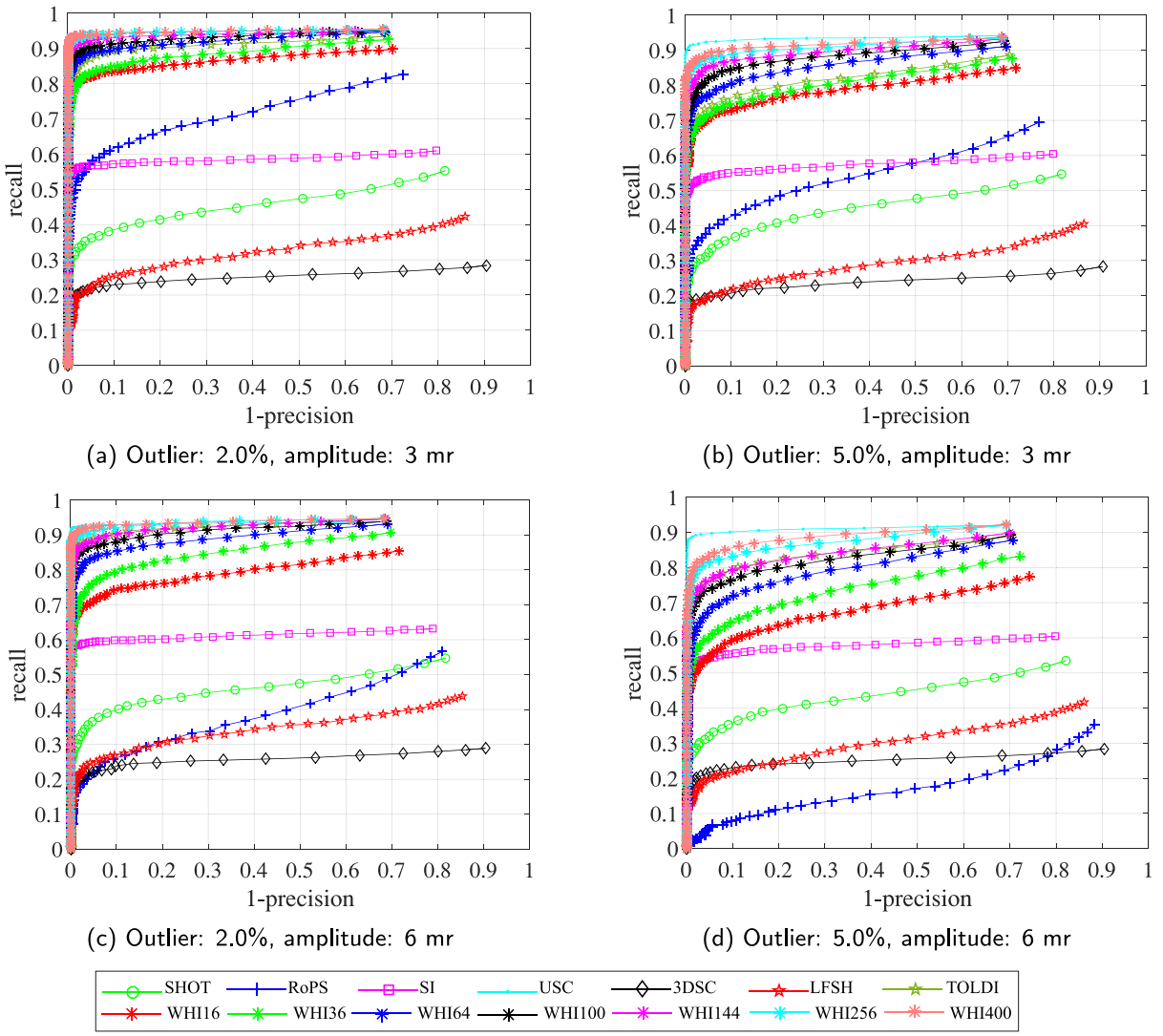

Fig. 12. Shot noises experiment: The descriptors are evaluated on the shot noises data. The figures show PRC curves with the shot noises of (a) 2.0% outlier and 3 mr amplitude, (b) 5.0% outlier and 3 mr amplitude, (c) 2.0% outlier and 6 mr amplitude and (d) 5.0% outlier and 6 mr amplitude, respectively.

In conclusion, the average value of the AUC_{pr} is used as the performance of accuracy. The overall accuracy evaluation result is shown in Fig. 14. The results show that the accuracy of WHI100, WHI144, WHI256 and WHI400 are higher than that of the USC. WHI256 is superior to the others. Although the TOLDI is also the method that encodes the 3D information in 2D forms, our 16-dimensional WHI achieves the AUC_{pr} performance of the 1200-dimensional TOLDI. Although WHI36 is close to the LFSH in dimension, WHI36 has 4.08 times the accuracy of the LFSH. All of the above experimental results fully illustrate the effectiveness of the proposed method.

4.5. Compactness

The dimension of the descriptors affects the memory footprint and the matching time. There is a trade-off needs to be made between the dimension and the accuracy. Researchers use the term compactness to represent the descriptiveness per unit floating number [14,50], defined as:

$$Compactness = \frac{\text{The average value of the } AUC_{pr}}{\text{The dimension of the descriptor}}. \tag{39}$$

According to Table 1, the compactness performance is calculated and shown in Fig. 15(a). The results in Fig. 15(a) show that WHI16, WHI36, WHI64 and WHI100 have higher compactness than the others. Fig. 15(b) shows the scatters of the AUC_{pr} versus dimension. Since the dimension of the LFSH is 30, the compactness is relatively higher than the others. However,

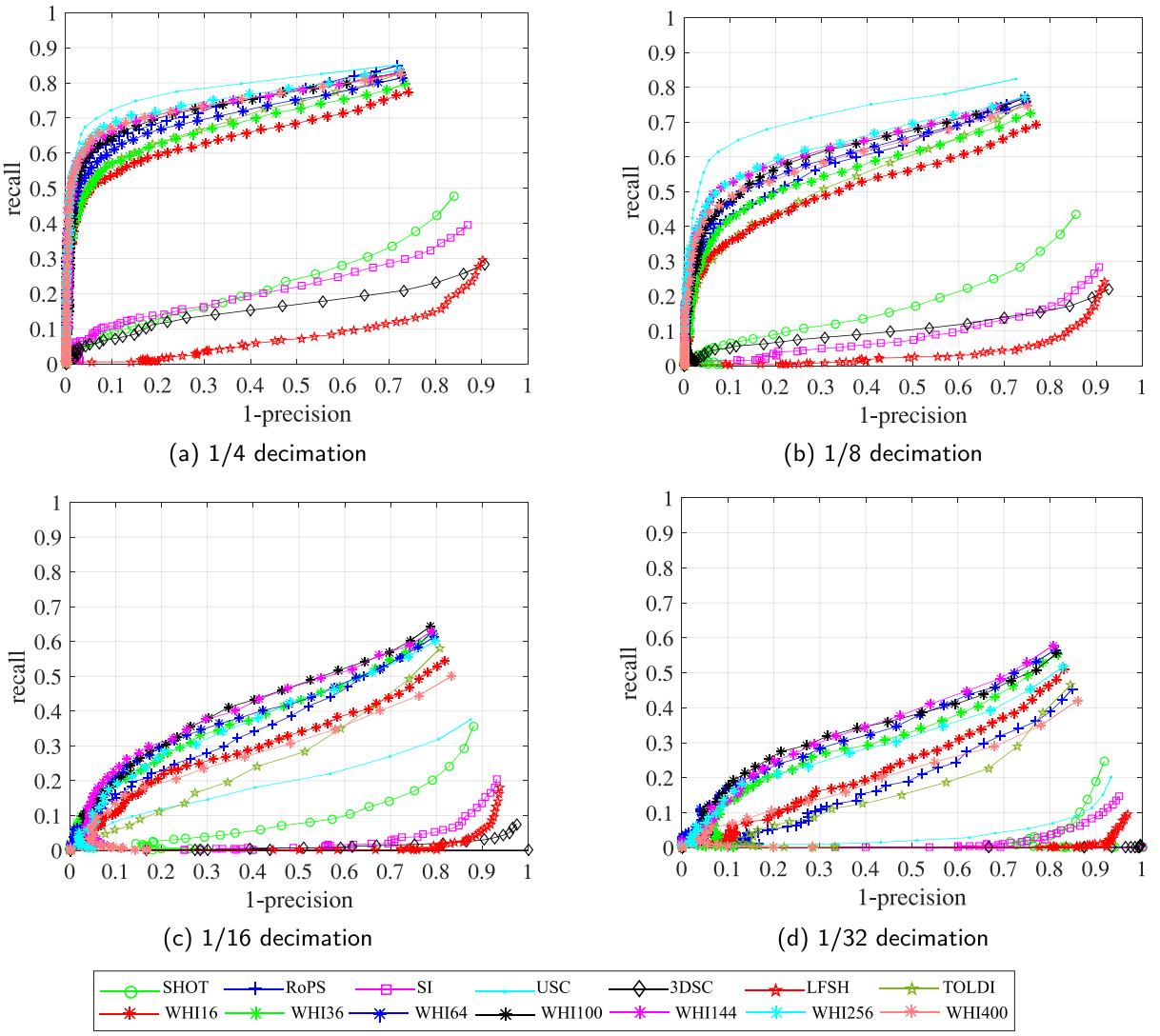

Fig. 13. Cross resolutions experiment: The scenes are downsampled to (a) 1/4 decimation, (b) 1/8 decimation, (c) 1/16 decimation and (d) 1/32 decimation, respectively. While the models keep the original density. The figures illustrate the PRC curves with cross resolution.

as recorded in Table 1, the accuracy of the WHI is much higher than that of the LFSH. As shown in Fig. 15(b), the 16-dimensional WHI16 is 3.67 times more accurate than the LFSH. Therefore, WHI16 achieves the highest compactness, which is 6.89 times as much as that of the LFSH. In terms of accuracy, WHI64 is very close to the USC, but about 30.33 times as compact as the USC. Furthermore, WHI16 has 103.26 times the compactness of the USC.

The results discussed in Section 4.4 show that the USC and TOLDI have higher robustness and descriptiveness. However, their compactness is severely restricted by their dimension. In contrast, the WHI requires only 64 dimensions to achieve the similar performance of the USC and 16 dimensions to achieve the equivalent performance of the TOLDI. Obviously, the descriptors based on the WHI are more compact. Although the LFSH achieves a high degree of compactness, the accuracy of the LFSH, as shown in Table 1 and Fig. 14, is relatively limited. As for our method, WHI16 not only has a fewer dimension than the LFSH, but also has a higher AUC_{pr} performance, which is an important breakthrough for the existing technologies. That is to say, the improvement of the compactness is free from sacrificing accuracy when the dimension decrease.

4.6. Efficiency

To evaluate the efficiency of each algorithm, we test the extraction time of each descriptor under the same condition. The computational time of an algorithm is often related to the number of points in the support region. So we change the number of points in the support region by controlling the support radius R from 5 mr to 30 mr. Then, we get the final result as shown in Fig. 16.

Table 1
The dimension of descriptors and the average AUC_{pr} on challenging datasets. The best results are shown in bold.

Descriptor	Dimension	Dataset (AUC_{pr})						Average AUC_{pr}
		Gaussian	Decimation	Gaussian+Decimation	Shot	Kinect	Space time	
WHI16	16	0.6975	0.4113	0.3230	0.7750	0.0765	0.2976	0.4302
WHI36	36	0.7275	0.4683	0.3821	0.8137	0.0971	0.3806	0.4782
WHI64	64	0.7494	0.4974	0.4027	0.8605	0.1064	0.4165	0.5054
WHI100	100	0.7651	0.5144	**0.4295**	0.8836	0.1113	0.4564	0.5267
WHI144	144	0.7706	**0.5214**	0.4182	0.8977	**0.1144**	0.4685	0.5318
WHI256	256	**0.7781**	0.5057	0.4014	0.9148	0.1137	0.4792	**0.5321**
WHI400	400	0.7731	0.4503	0.3704	0.9210	0.1124	**0.4902**	0.5196
HI400	400	0.7484	0.3928	0.3504	0.9058	0.0836	0.4579	0.4898
LRF(SHOT) +WHI400	400	0.7746	0.4514	0.3614	0.9186	0.1065	0.4817	0.5157
LRF(TOLDI) +WHI400	400	0.6455	0.3521	0.2716	0.8746	0.0677	0.4335	0.4408
SHOT	352	0.2915	0.1212	0.1063	0.4467	0.0889	0.4793	0.2557
RoPS	135	0.5625	0.4524	0.1976	0.4456	0.0810	0.3847	0.3540
SI	153	0.2469	0.0827	0.0240	0.5863	0.0052	0.2607	0.2010
USC	1960	0.7748	0.4274	0.3447	**0.9315**	0.1026	0.4808	0.5103
3DSC	1980	0.1342	0.0683	0.0468	0.2496	0.0622	0.0557	0.1028
LFSH	30	0.1162	0.0300	0.0096	0.3106	0.0098	0.2264	0.1171
TOLDI	1200	0.6343	0.4002	0.2864	0.8691	0.0685	0.3178	0.4294
W+TOLDI	1200	0.6437	0.4034	0.2891	0.8725	0.0717	0.3258	0.4344

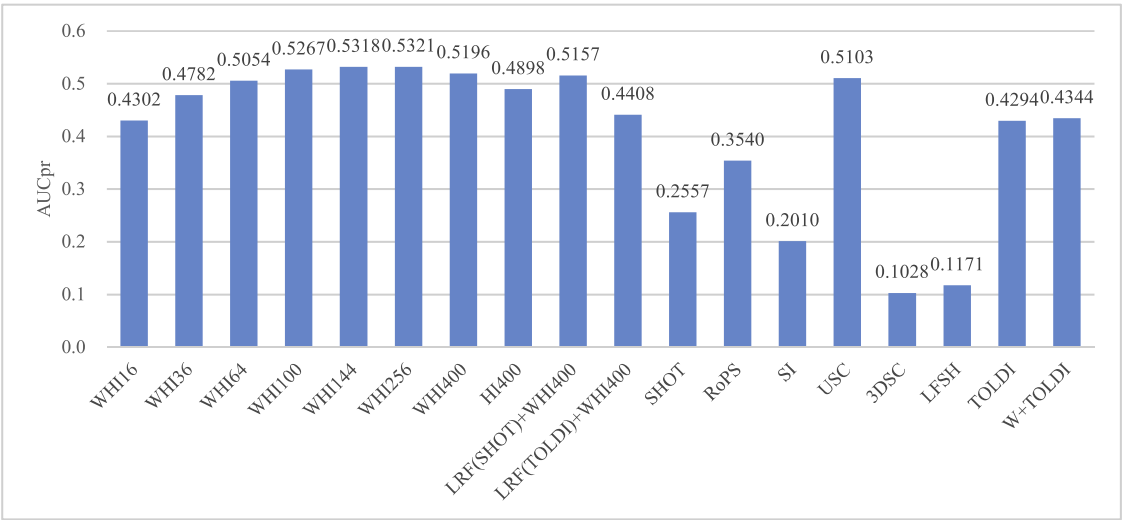

Fig. 14. Accuracy comparison: The average AUC_{pr} performance of descriptors.

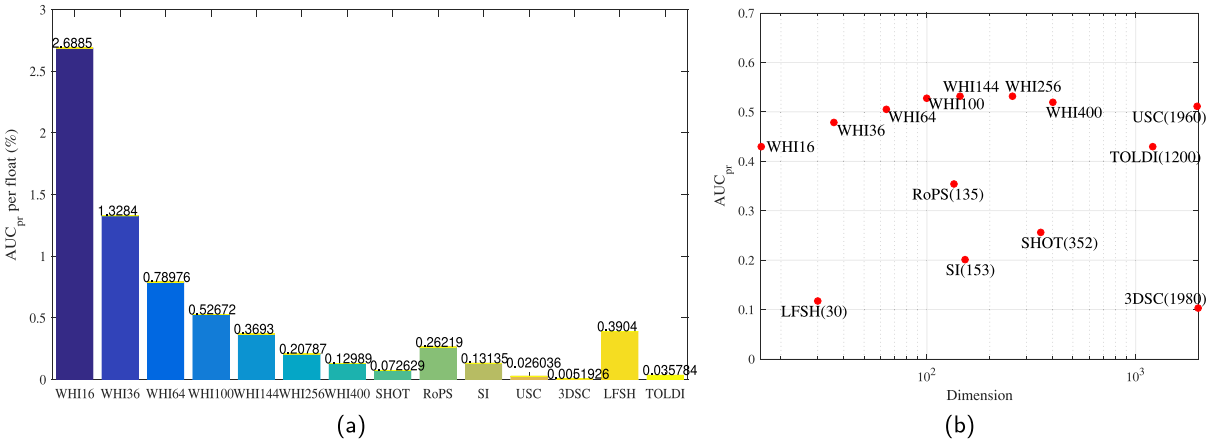

Fig. 15. The compactness performance of the descriptors: (a) Compactness histogram. (b) AUC_{pr} and dimension of descriptors. The *X*-axis is shown logarithmically for a better view.

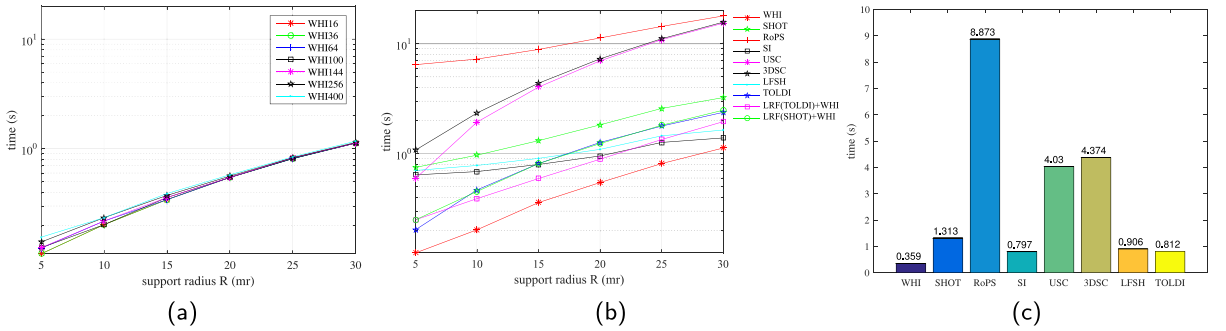

Fig. 16. Efficiency evaluation: (a) Computational time of the WHI with different dimension. (b) Computational time of the descriptors under different support radii. The *Y*-axis is shown logarithmically for a better view. (c) Computational time of the descriptors with the support radius of 15 mr.

According to the result shown in Fig. 16(a), the dimension of the WHI has very little effect on the computational time. Therefore, we use WHI100 to represent a series of WHIs in the comparison experiments. Fig. 16(b) illustrates that the WHI runs faster than the others, while the RoPS, USC and 3DSC are relatively time-consuming. When R is smaller, the TOLDI is faster. However, when R is larger than 15 mr, its efficiency will be lower than that of the SI and LFSH. Furthermore, we test the time of the WHI equipped with different LRFs that proposed in [36,42]. The results show that the LRF(SHOT)+WHI and LRF(TOLDI)+WHI are less efficient than our simplified LRF based method, which verifies the high efficiency of our simplified LRF. Generally, for the sake of efficiency, the support radius is usually set to 15 mr. The corresponding computational time of 15 mr is shown in Fig. 16(c). When the support radius is set to 15 mr, the WHI only takes 0.359 s, several times faster than the others. The high efficiency of the WHI is mainly attributed to the simplified LRF and the reuse of the calculations.

Remark. In this paper, we indirectly evaluate the performance of the proposed simplified LRF through the evaluation of the WHI descriptor. The results analyzed in Sections 4.4.4 and 4.5 can indirectly indicate the robustness and repeatability of the simplified LRF. Furthermore, the experiment fully proves the significant positive effect of the simplified LRF on the efficiency of the descriptor.

5. Point cloud registration application

An important application of the 3D local descriptors is point cloud registration that shares the pipelines with many applications, such as 3D reconstruction [33,38], 3D recognition [13,15,22] and surface alignment [8,50]. The point cloud registration is the transformation of the point clouds from different coordinate systems to the same coordinate system. The process can be formulated as:

$$(\mathbf{R}^*, \mathbf{T}^*) = \arg\min_{\mathbf{R},\mathbf{T}} \sum \|\mathbf{R}\cdot\mathbf{P}_S + \mathbf{T} - \mathbf{P}_T\|_2, \tag{40}$$

where $\mathbf{R}$ is the rotation matrix, $\mathbf{T}$ is the translation vector, $\mathbf{P}_S$ is the source point cloud and $\mathbf{P}_T$ is the target point cloud.

Usually, the descriptors are used as features for finding matching pairs between two different viewed point clouds. Then, the transformation between the two frames can be estimated based on these correspondences. In this paper, we propose a WHI based point cloud registration method, including two steps, i.e., the 3D local descriptor based coarse registration and the Iterative Closest Point (ICP) [2] based fine registration. In the coarse process, we use the WHI to find correspondences and estimate a coarse transformation. In the refined process, we use the popular ICP algorithm to refine the coarse result. Specifically, the ICP algorithm relies heavily on a good initialization provided by the coarse step. Meanwhile, this is the most time-consuming part of the registration phase [12]. In this section, we utilize WHI16 for the coarse point cloud registration. The support radius is set to 15 mr.

In this experiment, we use the public Stanford 3D Scanning Repository[2] dataset [5] for registration. The dataset contains four models ("Standford Bunny", "Armadillo", "Dragon" and "Happy Buddha"), as shown in Fig. 18. The number of points in the raw input is too large, which is time-consuming for the point feature extraction, matching and ICP in the registration. So, as the strategy in the paper [40], we downsample $\mathbf{P}_S$ and $\mathbf{P}_T$, getting $\mathbf{P}'_S$ and $\mathbf{P}'_T$. Then, the WHI features of $\mathbf{P}'_S$ and $\mathbf{P}'_T$ are extracted, denoted by $\mathbf{F}_S$ and $\mathbf{F}_T$, respectively. In order to generate matching pairs, we find the corresponding nearest feature in $\mathbf{F}_T$ for every feature of $\mathbf{F}_S$. The matching phase is implemented by using the kd-tree algorithm in FLANN [28] library. However, there may be false matches after the matching phase. So, RANdom SAmple Consensus (RANSAC) [9] algorithm is used for filtering out the false matching. We use all the inliers to estimate the rotation matrix $\mathbf{R}_C$ and the translation vector $\mathbf{T}_C$ by using Singular Value Decomposition (SVD). Then, we can get the coarse registration results $\mathbf{P}'_{SC}$ by transforming the source point cloud $\mathbf{P}'_S$ to the new coordinate system. Finally, the ICP algorithm is used for the fine registration. In order to

[2] http://graphics.stanford.edu/data/3Dscanrep/.

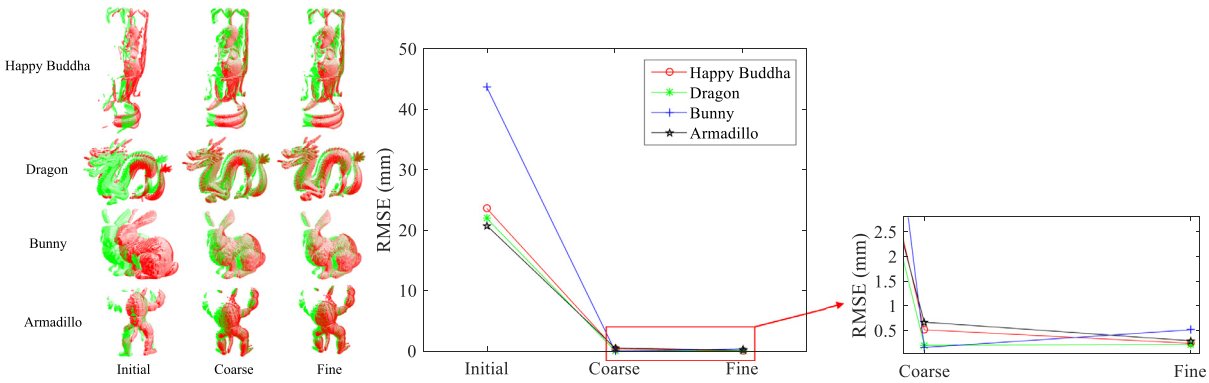

Fig. 17. The accuracy of the registration based on WHI16.

improve the efficiency of this phase, we take $\mathbf{P}'_{SC}$ and $\mathbf{P}'_{T}$ as inputs. The RMSE is used for evaluating the accuracy of the registration, defined as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \left| \mathbf{R}_{GT} \cdot \mathbf{p}_S^i + \mathbf{T}_{GT} - \mathbf{p}_o^i \right|^2}{N}} \tag{41}$$

where N is the total number of the source points, $\mathbf{R}_{GT}$ and $\mathbf{T}_{GT}$ are the ground-truth, $\mathbf{p}_S^i$ and $\mathbf{p}_o^i$ are the points of the source point cloud and the output registered point cloud, respectively.

Fig. 17 shows the registration results and RMSE performance. We set the downsampling leaf size to 2 mm. Fig. 17(b) shows that the results of the coarse registration are very close to that of the refined so that the fine registration can no longer continue to converge. This means that the coarse registration based on WHI16 can achieve the level of the fine registration. Fig. 18 shows the results of the coarse and fine registration on four models with the leaf size of 4 mm. We can see that it is difficult for the naked eyes to distinguish the coarse results from the refined ones.

Since we use a coarse-to-fine strategy, we can improve the speed by sacrificing the precision of the coarse registration to some extent while keeping a satisfying refined result. In Fig. 17, the coarse registration achieves higher accuracy, with the cost of higher point cloud density and more feature extraction time. To balance the coarse and fine registration to further speed up the algorithm, we increase the sampling rate to reduce the input points and registration time. In Table 2, under different leaf sizes, the accuracy and time are compared. The results show that with the increase of sampling rate, the efficiency of the registration is obviously improved. When the leaf size is 2 mm, the registration time is approximately 1.0 s. When the leaf size is 4 mm, although the accuracy of the coarse registration decreases, the RMSE of the fine registration does not increase significantly. When the leaf size increases to 7 mm, the registration time decreases to approximately 0.1 s, which is basically real-time, improved by approximately 10 times. At this time, the RMSE still maintains the same level

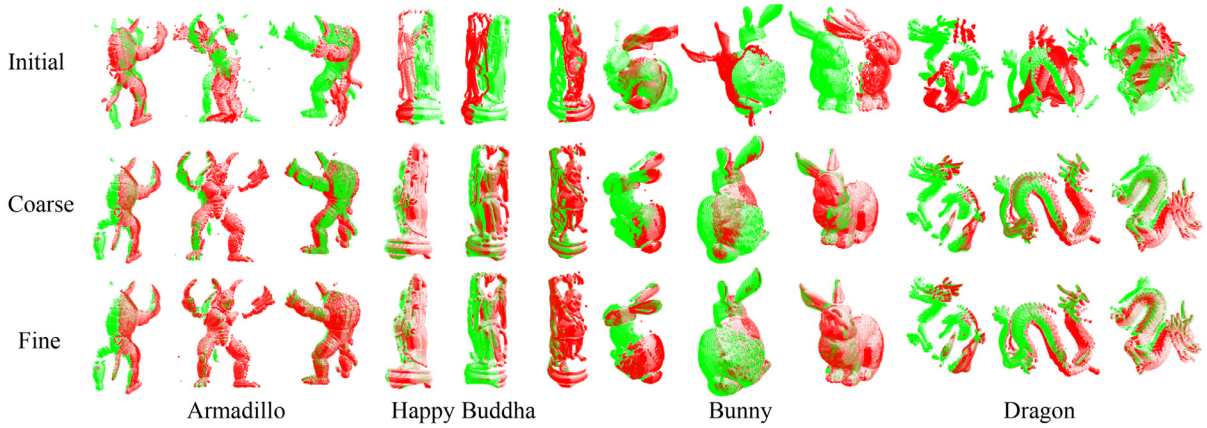

Fig. 18. The results of registration based on WHI16. From top to bottom are initial position, coarse results and fine results, respectively. The red and the green colors represent source and target point clouds, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 2
Registration time statistics and RMSE performance. (Num_s and Num_t represent the number of input source and target points, respectively. $Sample_s$ and $Sample_t$ represent the number of source and target points after downsampling, respectively. t_{ft}, t_{fm}, t_{ran}, t_{icp} and t_{sum} represent the feature extraction time, feature matching time, RANSAC time, ICP time and total registration time, respectively. $RMSE_c$ and $RMSE_f$ represent the RMSE performance of coarse and fine results, respectively).

Data	Happy Buddha			Dragon			Bunny			Armadillo		
leaf size (mm)	2	4	7	2	4	7	2	4	7	2	4	7
Num_s	75582			41841			40256			28220		
Num_t	69158			34836			40097			27315		
$Sample_s$	5304	1592	562	7155	2155	835	7133	2067	739	5243	1752	590
$Sample_t$	4633	1341	503	6312	1958	763	6813	1997	718	4935	1488	558
t_{ft} (s)	0.478	0.12	0.041	0.626	0.175	0.063	0.777	0.202	0.062	0.453	0.127	0.043
t_{fm} (s)	0.072	0.019	0.007	0.102	0.028	0.011	0.105	0.028	0.014	0.066	0.02	0.007
t_{ran} (s)	0.099	0.034	0.017	0.131	0.048	0.022	0.144	0.046	0.02	0.093	0.035	0.018
t_{icp} (s)	0.101	0.028	0.03	0.085	0.015	0.01	0.1	0.049	0.012	0.057	0.03	0.003
t_{sum} (s)	0.757	0.206	0.1	0.948	0.27	0.11	1.133	0.33	0.112	0.674	0.216	0.075
$RMSE_c$ (mm)	0.660	1.026	6.287	0.226	0.506	0.738	0.183	0.522	5.487	0.417	2.676	1.679
$RMSE_f$ (mm)	0.271	0.226	0.371	0.245	0.203	0.651	0.536	0.621	0.787	0.317	0.215	0.813

of magnitude compared with the leaf size of 2 mm and 4 mm. This is because the provided initial position is still valid for the ICP, even though the accuracy of the coarse registration has decreased.

6. Conclusion and future work

In this paper, we propose a novel WHI 3D descriptor based on a simplified LRF and the weighted height image. Experiments show that the proposed method is more robust, descriptive, efficient and compact than SOTA algorithms. The improvement is attributed to our simplified LRF, sharing computational results, and well-conditioned coding function.

On one hand, through theoretical analysis, we simplify the LRF and the 3D information coding by reducing and sharing calculations, which makes our method more efficient. On the other hand, theoretical analysis shows that the feature space of the 3D information is a well-conditioned problem, so we propose the WHI to make the feature space close to well-condition. The evaluation of the accuracy shows that our WHI has higher descriptiveness and robustness on the Gaussian noise, shot noises, decimation data and the occluded Space time and Kinect datasets. For the compactness, WHI16 is superior to the others due to the compact 2D expression. The results of the computational time experiment show that our method is several times more efficient than the others thanks to the simplified LRF and fast 3D information coding. Finally, based on the WHI, we propose a coarse-to-fine point cloud registration, which can achieve high accuracy and real-time performance. The experiments of point cloud registration further prove the high efficiency and accuracy of the WHI. The results also show that the WHI could be used in real-time applications.

In the future, the 3D information coding with the color and the mesh in the descriptor is an interesting research direction. With the development of the point cloud acquisition devices, such information is easy to obtain, and they have not been popularly utilized. Applying the WHI to more related projects is also worthwhile. Moreover, the global coding with local information is another interesting direction for scene understanding and object recognition.

Declaration of Competing Interest

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

CRediT authorship contribution statement

Tiecheng Sun: Conceptualization, Methodology, Software, Investigation, Data curation, Formal analysis, Writing - original draft. **Guanghui Liu:** Conceptualization, Formal analysis, Writing - original draft, Funding acquisition, Supervision. **Shuaicheng Liu:** Conceptualization, Formal analysis, Investigation, Funding acquisition, Writing - original draft. **Fanman Meng:** Conceptualization, Formal analysis, Funding acquisition, Visualization, Writing - original draft. **Liaoyuan Zeng:** Conceptualization, Formal analysis, Writing - original draft. **Ru Li:** Conceptualization, Formal analysis, Writing - original draft.

Acknowledgement

The authors would like to acknowledge Stanford University for providing the 3D registration models, Bologna University for providing the 3D scenes, PCL library and FLANN library. We thank Yulan Guo from National University of Defence Technology, Jiaqi Yang from Huazhong University of Science and Technology and Samuele Salti from University of Bologna for providing the source code. This work is supported in part by National Natural Science Foundation of China (NSFC) under

Grants 61872067, 61871087 and 61720106004, in part by Department of Science and Technology of Sichuan Province under Grant 2019YFH0016 and 2019YJ0166.

References

[1] J. Assfalg, M. Bertini, A. Del Bimbo, P. Pala, Content-based retrieval of 3-d objects using spin image signatures, IEEE Trans. Multimed. 9 (3) (2007) 589–599, doi:10.1109/TMM.2006.886271.
[2] P.J. Besl, N.D. McKay, A method for registration of 3-d shapes, IEEE Trans. Pattern Anal. Mach. Intell. 14 (2) (1992) 239–256, doi:10.1109/34.121791.
[3] R. Bro, E. Acar, T.G. Kolda, Resolving the sign ambiguity in the singular value decomposition, J. Chemometr. 22 (2) (2008) 135–140, doi:10.1002/cem.1122.
[4] J. Chen, Y. Fang, Y.K. Cho, Performance evaluation of 3d descriptors for object recognition in construction applications, Autom. Constr. 86 (2018) 44–52, doi:10.1016/j.autcon.2017.10.033.
[5] B. Curless, M. Levoy, A volumetric method for building complex models from range images, in: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, ACM, 1996, pp. 303–312, doi:10.1145/237170.237269.
[6] J. Davis, M. Goadrich, The relationship between precision-recall and ROC curves, in: Proceedings of the 23rd International Conference on Machine Learning (ICML), ACM, 2006, pp. 233–240, doi:10.1145/1143844.1143874.
[7] G. Elbaz, T. Avraham, A. Fischer, 3d point cloud registration for localization using a deep neural network auto-encoder, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 4631–4640, doi:10.1109/CVPR.2017.265.
[8] X. Fengguang, H. Xie, A 3d surface matching method using keypoint-based covariance matrix descriptors, IEEE Access 5 (2017) 14204–14220, doi:10.1109/ACCESS.2017.2727066.
[9] M.A. Fischler, R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Commun. ACM 24 (6) (1981) 381–395, doi:10.1145/358669.358692.
[10] A. Flint, A. Dick, A. Van Den Hengel, Thrift: local 3d structure recognition, in: 9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications (DICTA), IEEE, 2007, pp. 182–188, doi:10.1109/DICTA.2007.4426794.
[11] A. Frome, D. Huber, R. Kolluri, T. Bülow, J. Malik, Recognizing objects in range data using regional point descriptors, in: European Conference on Computer Vision (ECCV), 2004, pp. 224–237, doi:10.1007/978-3-540-24672-5_18.
[12] A. Gressin, C. Mallet, J. Demantké, N. David, Towards 3d lidar point cloud registration improvement using optimal neighborhood knowledge, ISPRS-J. Photogramm. Remote 79 (2013) 240–251, doi:10.1016/j.isprsjprs.2013.02.019.
[13] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, 3d object recognition in cluttered scenes with local surface features: a survey, IEEE Trans. Pattern Anal. Mach. Intell. 36 (11) (2014) 2270–2287, doi:10.1109/TPAMI.2014.2316828.
[14] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, N.M. Kwok, A comprehensive performance evaluation of 3d local feature descriptors, Int. J. Comput. Vis. 116 (1) (2016) 66–89, doi:10.1007/s11263-015-0824-y.
[15] Y. Guo, F. Sohel, M. Bennamoun, M. Lu, J. Wan, Rotational projection statistics for 3d local surface description and object recognition, Int. J. Comput. Vis. 105 (1) (2013) 63–86, doi:10.1007/s11263-013-0627-y.
[16] Y. Guo, F. Sohel, M. Bennamoun, J. Wan, M. Lu, An accurate and robust range image registration algorithm for 3d object modeling, IEEE Trans. Multimed. 16 (5) (2014) 1377–1390, doi:10.1109/TMM.2014.2316145.
[17] Y. Guo, F.A. Sohel, M. Bennamoun, J. Wan, M. Lu, ROPs: a local feature descriptor for 3d rigid objects based on rotational projection statistics, in: International Conference on Communications, Signal Processing, and their Applications (ICCSPA), 2013, pp. 1–6, doi:10.1109/ICCSPA.2013.6487310.
[18] Y. He, Y. Mei, An efficient registration algorithm based on spin image for lidar 3d point cloud models, Neurocomputing 151 (2015) 354–363, doi:10.1016/j.neucom.2014.09.029.
[19] C. Hong, J. Yu, D. Tao, M. Wang, Image-based three-dimensional human pose recovery by multiview locality-sensitive sparse retrieval, IEEE Trans. Ind. Electron. 62 (6) (2015) 3742–3751, doi:10.1109/TIE.2014.2378735.
[20] C. Hong, J. Yu, J. You, X. Chen, D. Tao, Multi-view ensemble manifold regularization for 3d object recognition, Inf. Sci. 320 (2015) 395–405, doi:10.1016/j.ins.2015.03.032.
[21] A.E. Johnson, M. Hebert, Using spin images for efficient object recognition in cluttered 3d scenes, IEEE Trans. Pattern Anal. Mach. Intell. 21 (5) (1999) 433–449, doi:10.1109/34.765655.
[22] O. Kechagias-Stamatis, N. Aouf, G. Gray, L. Chermak, M. Richardson, F. Oudyi, Local feature based automatic target recognition for future 3d active homing seeker missiles, Aerosp. Sci. Technol. 73 (2018) 309–317, doi:10.1016/j.ast.2017.12.011.
[23] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2) (2004) 91–110, doi:10.1023/B:VISI.0000029664.99615.94.
[24] S. Malassiotis, M.G. Strintzis, Snapshots: a novel local surface descriptor and matching algorithm for robust 3d surface alignment, IEEE Trans. Pattern Anal. Mach. Intell. 29 (7) (2007) 1285–1290, doi:10.1109/TPAMI.2007.1060.
[25] T. Masuda, Log-polar height maps for multiple range image registration, Comput. Vis. Image Underst. 113 (11) (2009) 1158–1169, doi:10.1016/j.cviu.2009.05.003.
[26] A. Mian, M. Bennamoun, R. Owens, On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes, Int. J. Comput. Vis. 89 (2-3) (2010) 348–361, doi:10.1007/s11263-009-0296-z.
[27] K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, IEEE Trans. Pattern Anal. Mach. Intell. 27 (10) (2005) 1615–1630, doi:10.1109/TPAMI.2005.188.
[28] M. Muja, D.G. Lowe, Fast approximate nearest neighbors with automatic algorithm configuration 2(331-340) (2009) 2. 10.1.1.160.1721
[29] A. Petrelli, L. Di Stefano, On the repeatability of the local reference frame for partial shape matching, in: IEEE International Conference on Computer Vision (ICCV), 2011, pp. 2244–2251, doi:10.1109/ICCV.2011.6126503.
[30] R.B. Rusu, N. Blodow, M. Beetz, Fast point feature histograms (FPFH) for 3d registration, in: IEEE International Conference on Robotics and Automation (ICRA), 2009, pp. 3212–3217, doi:10.1109/ROBOT.2009.5152473.
[31] R.B. Rusu, N. Blodow, Z.C. Marton, M. Beetz, Aligning point cloud views using persistent feature histograms, in: IEEE International Conference on Intelligent Robots and Systems, 2008, pp. 3384–3391, doi:10.1109/IROS.2008.4650967.
[32] R.B. Rusu, S. Cousins, 3d is here: point cloud library (PCL), in: IEEE International Conference on Robotics and Automation (ICRA), 2011, pp. 1–4, doi:10.1109/ICRA.2011.5980567.
[33] S. Salti, F. Tombari, L. Di Stefano, Shot: unique signatures of histograms for surface and texture description, Comput. Vis. Image Underst. 125 (2014) 251–264, doi:10.1016/j.cviu.2014.04.011.
[34] T. Sun, S. Liu, G. Liu, S. Zhu, Z. Zhu, A 3d descriptor based on local height image, in: 2018 IEEE International Symposium on Circuits and Systems (ISCAS), IEEE, 2018, pp. 1–5, doi:10.1109/ISCAS.2018.8351336.
[35] F. Tombari, S. Salti, L. Di Stefano, Unique shape context for 3d data description, in: Proceedings of the ACM Workshop on 3D Object Retrieval, 2010, pp. 57–62, doi:10.1145/1877808.1877821.
[36] F. Tombari, S. Salti, L. Di Stefano, Unique signatures of histograms for local surface description, in: European Conference on Computer Vision (ECCV), 2010, pp. 356–369, doi:10.1007/978-3-642-15558-1_26.
[37] F. Tombari, S. Salti, L. Di Stefano, Performance evaluation of 3d keypoint detectors, Int. J. Comput. Vis. 102 (1-3) (2013) 198–220, doi:10.1007/s11263-012-0545-4.
[38] Y. Xu, S. Tuttas, L. Hoegner, U. Stilla, Reconstruction of scaffolds from a photogrammetric point cloud of construction sites using a novel 3d local feature descriptor, Autom. Constr. 85 (2018) 76–95, doi:10.1016/j.autcon.2017.09.014.

[39] B. Yang, Z. Wei, Q. Li, J. Li, Automated extraction of street-scene objects from mobile lidar point clouds, Int. J. Remote Sens. 33 (18) (2012) 5839–5861, doi:10.1080/01431161.2012.674229.
[40] J. Yang, Z. Cao, Q. Zhang, A fast and robust local descriptor for 3d point cloud registration, Inf. Sci. 346 (2016) 163–179, doi:10.1016/j.ins.2016.01.095.
[41] J. Yang, Y. Xiao, Z. Cao, Toward the repeatability and robustness of the local reference frame for 3d shape matching: an evaluation, IEEE Trans. Image Process. 27 (8) (2018) 3766–3781, doi:10.1109/TIP.2018.2827330.
[42] J. Yang, Q. Zhang, Y. Xiao, Z. Cao, Toldi: an effective and robust approach for 3d local shape description, Pattern Recognit. 65 (2017) 175–187, doi:10.1016/j.patcog.2016.11.019.
[43] J. Yu, Z. Kuang, B. Zhang, W. Zhang, D. Lin, J. Fan, Leveraging content sensitiveness and user trustworthiness to recommend fine-grained privacy settings for social image sharing, IEEE Trans. Inf. Forensic Secur. 13 (5) (2018) 1317–1332, doi:10.1109/TIFS.2017.2787986.
[44] J. Yu, Y. Rui, D. Tao, Click prediction for web image reranking using multimodal sparse coding, IEEE Trans. Image Process. 23 (5) (2014) 2019–2032, doi:10.1109/TIP.2014.2311377.
[45] J. Yu, M. Tan, H. Zhang, D. Tao, Y. Rui, Hierarchical deep click feature prediction for fine-grained image recognition, IEEE Trans. Pattern Anal. Mach. Intell. (2019), doi:10.1109/TPAMI.2019.2932058.
[46] J. Yu, D. Tao, M. Wang, Y. Rui, Learning to rank using user clicks and visual features for image retrieval, IEEE Trans. Cybern. 45 (4) (2014) 767–779, doi:10.1109/TCYB.2014.2336697.
[47] J. Yu, C. Zhu, J. Zhang, Q. Huang, D. Tao, Spatial pyramid-enhanced netvlad with weighted triplet loss for place recognition, IEEE Trans. Neural Netw. Learn. Syst. (2019), doi:10.1109/TNNLS.2019.2908982.
[48] B. Zhao, X. Le, J. Xi, A novel SDASS descriptor for fully encoding the information of a 3d local surface, Inf. Sci. 483 (2019) 363–382, doi:10.1016/j.ins.2019.01.045.
[49] B. Zhao, J. Xi, Efficient and accurate 3d modeling based on a novel local feature descriptor, Inf. Sci. (2019), doi:10.1016/j.ins.2019.04.020.
[50] Y. Zou, X. Wang, T. Zhang, B. Liang, J. Song, H. Liu, Broph: an efficient and compact binary descriptor for 3d point clouds, Pattern Recognit. 76 (2018) 522–536, doi:10.1016/j.patcog.2017.11.029.