Information Sciences xxx (2016) xxx–xxx

Contents lists available at ScienceDirect

Information Sciences

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

A fast and robust local descriptor for 3D point cloud registration

Q1 Jiaqi Yang, Zhiguo Cao, Qian Zhang*

Q2 *National Key Laboratory of Science and Technology on Multi-spectral Information Processing, School of Automation, Huazhong University of Science and Technology, China*

ARTICLE INFO

Article history:
Received 8 June 2015
Revised 21 January 2016
Accepted 30 January 2016
Available online xxx

Keywords:
Local feature descriptor
3D point cloud registration
Point correspondences
Self-similar models
Feature matching

ABSTRACT

This paper proposes a novel local feature descriptor, called a local feature statistics histogram (LFSH), for efficient 3D point cloud registration. An LFSH forms a comprehensive description of local shape geometries by encoding their statistical properties on local depth, point density, and angles between normals. The sub-features in the LFSH descriptor are low-dimensional and quite efficient to compute. In addition, an optimized sample consensus (OSAC) algorithm is developed to iteratively estimate the optimum transformation from point correspondences. OSAC can handle the challenging cases of matching highly self-similar models. Based on the proposed LFSH and OSAC, a coarse-to-fine algorithm can be formed for 3D point cloud registration. Experiments and comparisons with the state-of-the-art descriptors demonstrate that LFSH is highly discriminative, robust, and significantly faster than other descriptors. Meanwhile, the proposed coarse-to-fine registration algorithm is demonstrated to be robust to common nuisances, including noise and varying point cloud resolutions, and can achieve high accuracy on both model data and scene data.

1. Introduction

Registration of point clouds captured by 3D scanner devices is a fundamental problem in 3D computer vision. Numerous applications have been developed, including 3D modeling [30,32], object recognition [3,25], pose estimation [21], face recognition [29], and surface alignment [1,18]. The aim is to transform the data acquired in various views into a common coordinate system, which is optimal for model recovery or pose estimation. It is a challenging task, owing to the following difficulties: unknown initial positions of the input data, noisy raw data, varying point cloud resolutions (point densities), and partial overlap between two models. Therefore, further studies are required to improve the performance of the registration methods in practice.

In general, point cloud registration methods can be classified into three categories: greedy searching-based [8,9], global feature-based [19,52], and local feature-based methods [10,24,26,36,44,50,51]. The greedy searching-based methods randomly select several points (usually three) from the source data and find the corresponding points from the target data based on an exhaustive search on the target. By considering all possible correspondences, these methods guarantee an accurate solution, but often give rise to a heavy computational load, especially for high-resolution data. Both global and local feature-based methods involve two steps: coarse and fine registration [48]. The coarse registration aims at estimating an

Q3 * Corresponding author. Tel.: +86 2787558918.
E-mail addresses: yjq_hust@163.com (J. Yang), zgcao@hust.edu.cn (Z. Cao), hangfanzq@163.com (Q. Zhang).

http://dx.doi.org/10.1016/j.ins.2016.01.095
0020-0255/

initial transformation between two point clouds. It is achieved by matching features and generating point-to-point correspondences. The resultant alignment is usually further refined using the iterative closest point (ICP) algorithm or its variants [6,13,37,43]. The necessity of coarse registration is demonstrated in two aspects. First, the ICP algorithms start with an initial guess for estimating the rigid transformation. If two point clouds are not spatially close, the ICP algorithms may get trapped in a local minimum [44]. Second, a good initial transformation can significantly improve the computational efficiency of the ICP algorithms [39]. The main difference between the two feature-based methods is the 3D features they use. Global features describe the entire model shape, whereas local features only encode the neighborhood characteristics of the feature points. Owing to the limitation that global features may not be fully contained inside the overlapped region, they therefore hold limited discriminative power. Compared with global features, local features are more suitable for aligning partially overlapped point clouds.

In local feature-based methods, local feature descriptors play a crucial role in feature matching [22,24]. In general, a good feature descriptor should be highly descriptive, in order to provide a comprehensive and unambiguous representation of local shape geometry. To ensure accurate and efficient feature matching, the feature descriptors should also be computationally efficient, compact, and robust to common nuisances such as noise and point cloud resolution variation. At present, numerous local feature descriptors have been proposed, including point signatures [10], spin images [26], fast point feature histograms (FPFH) [44], signature of histograms of orientations (SHOT) [51], and rotational projection statistics (RoPS) [23]. Readers may refer to a recent survey [20] for more details. These local feature descriptors can be categorized according to their use of local reference frames (LRFs) [23]. Feature descriptors without LRFs usually calculate the statistics of the local geometric information, e.g., normal, curvature. They are easy to compute, but always suffer from low descriptiveness [45]. In contrast, feature descriptors with LRFs, as shown in [23,26], encode the spatial distribution of the neighboring points with respect to the defined LRF [23]. Most LRF-based descriptors, such as spin images and RoPS, provide a fairly detailed description of local shapes. As a result, they have high-dimensional feature vectors, and require additional computation time. Another challenging problem is that constructing a unique LRF for almost planar or spherical surfaces is difficult. In addition, many LRF-based descriptors are not robust to surface noise and single outliers [35]. However, feature descriptors without LRFs perform better in such circumstances. Consequently, efficiency and robust description are very important in the design of local descriptors for 3D point cloud registration.

In the literature, there are many effective representations for encoding local shape geometry, including local depth [10,33] and point density [23,27]. Most existing efforts focus on describing the local shape geometry from a single aspect using 2D or 1D representations (such as spin images or FPFH), and therefore suffer from limited descriptiveness and/or poor robustness. In this context, we conduct a study on analyzing and encoding some typical local geometric features and integrating them into a new descriptor. Specifically, we employ the geometric features brought by the local depth, point density, and deviation angles between normals, to create an effective and efficient local shape description. As a new paradigm, multi-view learning introduces one function to model a particular view, and jointly optimizes all the functions to combine all multiple source features into a single feature [53,54]. It has been receiving increased attention recently. However, these discriminative features ideally occur often enough in the 3D data to be learned. Therefore, we directly concatenate the three sub-features into a new descriptor in our method, which appears to be a popular operation as in [23,51], and then apply it to 3D point cloud registration.

The illustration of the proposed registration algorithm is shown in Fig. 1. First, two given point clouds, namely the source and target data discussed in the remainder of this paper, are simplified for efficient feature extraction. Then, we present the LFSH descriptor for local shape description. The LFSH descriptor employs local geometric information from several aspects, i.e., local depth, point density, and deviation angles between normals. For each aspect, feature statistics are computed with respect to a local reference axis (LRA). By combining these feature statistics in a histogram, we can obtain the LFSH descriptor. In this paper, the normal of the query point is taken as the LRA. The repeatability of normals under rigid transformation has been demonstrated in [40]. It promises that features extracted with respect to normals are invariant to an object's pose. Moreover, compared to LRF, the defined LRA can be constructed rapidly and is robust to many nuisances, including noise and varying point cloud resolutions [38]. The developed LFSH descriptor achieves a fast and robust description for local geometric information, as will be shown in Section 3. Next, correspondences between two point clouds are established via feature matching. An optimized sample consensus (OSAC) algorithm is presented for estimating an optimum rigid transformation from the correspondences. An OSAC includes a novel error metric that measures the correctness of a rigid transformation. It is calculated for two point clouds rather than point correspondences. Finally, the transformation is further refined and used to transform the source data to the target data. The contributions of this work are as follows:

(1) A novel local feature descriptor named LFSH is proposed. It is significantly fast for computation yet quite distinctive.
(2) We introduce an optimized sample consensus (OSAC) algorithm to overcome the challenging matching problem for highly self-similar models.
(3) A coarse-to-fine registration algorithm for 3D point clouds is presented; the algorithm is fast, precise, and can be used in real-time applications.

The remainder of this paper is organized as follows. The following section (Section 2) presents a brief review of related work on local feature descriptors. Section 3 introduces the proposed LFSH feature descriptor. Section 4 presents the details of the coarse-to-fine registration algorithm. In Section 5, experimental results and comparisons are given to validate the effectiveness and efficiency of the proposed approach. Conclusions and future work are discussed in Section 6.

Please cite this article as: J. Yang et al., A fast and robust local descriptor for 3D point cloud registration, Information Sciences (2016), http://dx.doi.org/10.1016/j.ins.2016.01.095

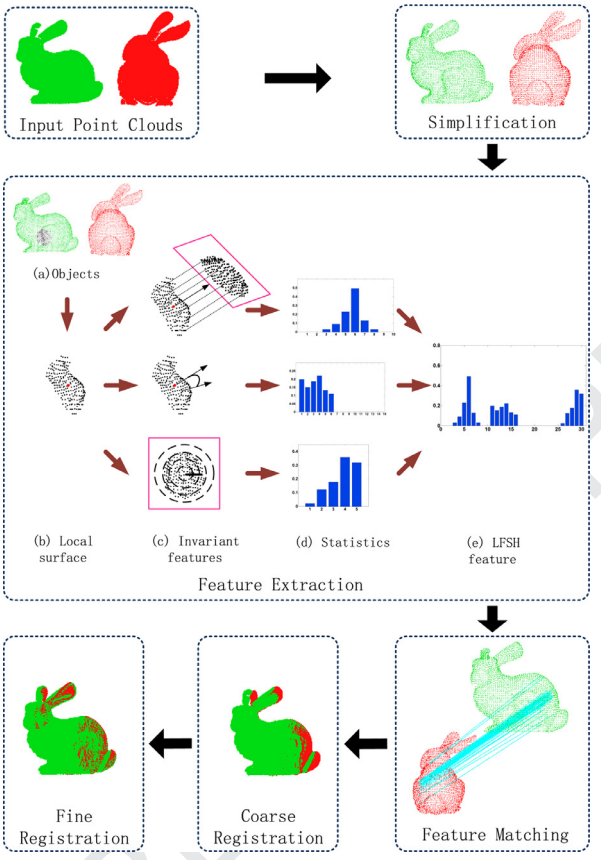

Fig. 1. Illustration of the proposed 3D point cloud registration algorithm. (For clarity, the source and target point clouds are respectively shown in red and green in the remainder of this paper.) (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

2. Related work

Numerous studies on point cloud registration have been conducted. In these methods, refined registration algorithms, such as ICP [6], have already been able to achieve fairly precise results. The real challenge for 3D point cloud registration lies in the coarse registration process. Because our method is a local feature-based approach, the most relevant methods for local shape description are presented in this section.

Stein and Medioni [49] proposed a splash descriptor by encoding the relationship (angular distance) between the query point and its geodesic neighbors using an LRF; the relationship was then stored using a 3D vector, which was finally transformed into curvatures and torsion angles. Chua and Javis [10] placed a sphere centered at the query point to obtain a contour by intersecting the surface. Subsequently, the distances from points on the contour to the tangent plane at the query point and the clockwise rotation angles from a reference direction were expressed by a set of discrete values, which formed the point signature descriptor. Because only neighbors on one contour are considered, the descriptiveness of these two descriptors is limited. Johnson and Hebert [26] proposed the spin image descriptor for 3D surface matching, which is robust to noise but expensive to compute, and sensitive to mesh resolution variation. They first specified the normal of the query point as the reference axis, and projected the neighbors defined by the support angle to a 2D coordinate; a gray image was then generated using a 2D array accumulator. Several variants of the spin image have been proposed to either reduce the time costs of feature extraction or improve its descriptiveness; examples include a face-based spin image [7], a spherical spin image [42], a multi-resolution spin image [11], and a spin image signature [4]. Similarly, “snapshots” [33] and log polar height map (LPHM) [34] descriptors are represented by 2D image arrays as well. Unlike the point density used in a spin image, these two descriptors mainly describe local depth information and have been proven to be more discriminative than the spin image. Sun and Abidi [50] generated geodesic circles around the query point and projected these circles onto the tangent plane of the query point, which formed the “point’s fingerprint” descriptor. Because more contours are generated, “point’s fingerprint” contains more discriminative information than the methods that only use one contour (e.g., point signature) or a 2D histogram (e.g., spin images).

Please cite this article as: J. Yang et al., A fast and robust local descriptor for 3D point cloud registration, Information Sciences (2016), http://dx.doi.org/10.1016/j.ins.2016.01.095

Later on, Frome et al. [16] proposed a 3D shape context (3DSC) descriptor, which was an extension of the 2D shape context method [5]. They divided the radius neighbors of the query point into bins and counted the weighted number of points in each bin. Gelfand et al. [17] proposed an integral volume descriptor; they calculated the volume of the intersection of a ball centered at the query point with the interior of the surface. The integral volume descriptor is low-dimensional and robust to noise; however, it has limited descriptiveness because only the surface volume feature is encoded. Flint et al. [14,15] introduced a THRIFT descriptor by extending the 2D scale invariant feature transform (SIFT) [31] descriptor to 3D space. They used the deviation angles between the surface normal at the key point and the surface normals at the neighboring points to characterize local shape geometry. The contribution of each neighboring point is determined by two factors: the density of point samples and the distance from the neighboring point to the key point. Rusu et al. [47] proposed a novel method to characterize the local geometry by using point feature histograms (PFH). PFH has great discriminative power and has been proven to be quite effective for point cloud registration in [45]. However, the calculation of PFH is quite expensive owing to its computational complexity of $O(k^2)$. In order to improve time efficiency, they subsequently used the simplified point feature histogram (SPFH) of neighbors to obtain the fast point feature histograms (FPFH) descriptor [44]. FPFH retains most of the discriminative power of PFH and is efficient to compute. Tombari et al. [51] first introduced a unique and unambiguous LRF and proposed the signature of histograms of orientations (SHOT). SHOT descriptor provides a very good balance between descriptiveness and time efficiency, but it is sensitive to mesh resolution variation. The method proposed by Guo et al. [23] to build the LRF is more robust compared with the method in [51]. They also extracted a rotational projection statistics (RoPS) descriptor by projecting the rotated neighbors onto three coordinate planes (i.e., the *xy*, *xz*, and *yz* planes) and calculating a set of feature statistics.

All these state-of-the-art local feature descriptors have made notable contributions to 3D point cloud registration. However, few of them achieve a good balance among descriptive ability, stability, and time efficiency, which are all crucial factors for 3D point cloud registration in practical applications.

3. Local feature statistics histograms

Local feature-based methods for 3D registration align models by using consistent point-to-point correspondences. They are usually generated via matching feature descriptors. Therefore, the feature descriptor should be distinctive and robust to various nuisances such as noise and varying point cloud resolution, in order to provide sufficient correct correspondences. In addition, the feature descriptor should be invariant to rigid transformation. In this section, we first introduce a fast and robust local feature descriptor named LFSH by computing the statistics of three local invariant features. Then, the LFSH generation parameters are analyzed.

3.1. Definition of LFSH

In this paper, we construct the LFSH descriptor by employing the local shape geometry from three different aspects: local depth, point density, and deviation angles between normals. Fig. 2 shows the illustration of the three local geometric features.

Assume that P is the input point cloud, which consists of N points $\{p_1, p_2, \ldots, p_N\}$. Given a query point p_i, a sphere of radius r is centered at p_i. It is used for determining the neighbors of p_i. The points inside the sphere excluding p_i are the radius neighbors of p_i, which are denoted by $P_n^i = \{p_i^j | j = 1, 2, \ldots, k\}$. See Fig. 2(a). Let n_i and n_i^j denote the normal vectors of p_i and p_i^j, respectively. Here, n_i is set as the LRA of the local surface that is corrupted by the sphere. Along the positive direction of LRA, a 2D plane L that is a tangent plane of the sphere is defined as the projection plane. Hence, all neighbors of p_i are projected on L, forming a new point set $P'^i_n = \{p'^j_i | j = 1, 2, \ldots, k\}$. Inspired by Malassiotis and Strintzis [33] and Masuda [34], we define the distance between p_i^j and p'^j_i as the local depth with a range of $[0, 2r]$. It is defined as

$$d_j = r - n_i \cdot (q_i^j - q_i), \tag{1}$$

where $\cdot$ represents a dot-product operation.

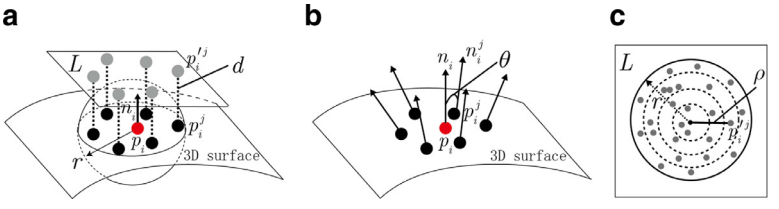

Fig. 2. Illustration of three local geometric features. (a) Local depth. (b) Deviation angle between normals. (c) Horizontal projection distance. Red, black, and gray points represent the query point, the radius neighbors, and the projected neighboring points, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Please cite this article as: J. Yang et al., A fast and robust local descriptor for 3D point cloud registration, Information Sciences (2016), http://dx.doi.org/10.1016/j.ins.2016.01.095

The second feature we used is the deviation angle between normals. See Fig. 2(b). To estimate the normals for 3D points, the method proposed in [38] is adopted in this paper. It has an obvious advantage of being robust to noise. A brief description is as follows. Given a point p_i in P, the radius neighbors of P_n^i are determined using the above method. Let $\overline{p_i}$ be the centroid of P_n^i. The 3×3 covariance matrix $Cov(p_i)$ is given by

$$Cov(p_i) = \begin{bmatrix} p_i^1 - \overline{p_i} \\ \cdots \\ p_i^k - \overline{p_i} \end{bmatrix}^T \cdot \begin{bmatrix} p_i^1 - \overline{p_i} \\ \cdots \\ p_i^k - \overline{p_i} \end{bmatrix}. \tag{2}$$

The eigenvector corresponding to the minimum eigenvalue of $Cov(p_i)$ is computed as the normal of p_i. Previous studies [14,44,47] have shown that a representation based on the deviation angle between a pair of normals has great discriminative power. Because the calculation of every pair of points in Q is quite expensive [44], we reduce the computational complexity to $O(k)$ (as opposed to $O(k^2)$ in [45]) by only calculating the deviation angles between n_i and n_i^j. This is a trade-off between the feature's descriptiveness and time efficiency. The angles can be represented as

$$\theta_j = \arccos(n_i \cdot n_i^j), \tag{3}$$

where $\theta_j \in [0, \pi]$.

Projecting 3D points onto a 2D plane provides a concise and efficient method to describe 3D local shape geometry [23]. A common representation for the 2D point distribution used in [23,26] is a 2D image. However, such encoding methods require a unique and stable LRF. Consequently, we propose a new feature to encode the 2D point distribution with respect to the LRA. It is capable of describing the distribution information. See Fig. 2(c). Here, we first place a circle of radius r at the crossing point of n_i and L. Next, the circle is divided into several annuli with the same width. Finally, a histogram is generated by counting the number of points that fall inside each annulus. The ratio of points distributed in each annulus can be calculated using the distance of $p_i'^j$ from the crossing point p'_i. Here, we name the distance as the horizontal projection distance ρ, which is defined as

$$\rho_j = \sqrt{\|p_i - p_i^j\|^2 - (n_i \cdot (p_i - p_i^j))^2}, \tag{4}$$

where $\|\cdot\|$ represents the distance between p_i and p_i^j. The range of ρ_j is $[0, r]$.

After three local geometric features have been presented, we can obtain three sub-histograms through the statistics of the three features using 1D array accumulators. The sub-histograms contain N_1, N_2, and N_3 bins, respectively. They are then further normalized such that the sum of all bins is equal to one, in order to achieve robustness to variations in point cloud resolution. By concatenating these three sub-histograms into one histogram, the LFSH descriptor is formed. The LFSH descriptor has three major characteristics, which are summarized as follows.

(1) The LFSH descriptor is quite efficient for calculation (as demonstrated in Section 5.1.3). The three local geometric features employed in the LFSH descriptor are low-dimensional and efficient to compute. The computational complexity of the LFSH descriptor is $O(k)$, where k is the size of the local surface.
(2) The LFSH descriptor is supposed to be robust for several reasons (as demonstrated in Section 5.1.2). First, the three employed geometric features are gained via 3D-2D projection (i.e., local depth and horizontal projection distance) or normal characterization (i.e., deviation angle between normals). The projection along a stable axis has been proved to be robust to noise and other nuisances [33]. The normal estimation method in [38] can provide the correct directions of normals in coarse sampled surfaces. It allows us a robust description for normal characteristics. Second, because the feature statistics in the LFSH descriptor have been normalized, variations in point cloud resolution would have a minor impact on it. Third, the three sub-features are all encoded with respect to a unique and stable LRA, which helps to boost the robustness of the associated features.
(3) Compared with the methods that describe local shape geometry from a single aspect (e.g., spin images [26] and THRIFT [14]), the LFSH descriptor offers a comprehensive representation for a local shape by employing it from several different aspects.

3.2. Analyzation of LFSH parameters

The descriptiveness of a descriptor is critical for establishing correct correspondences between point clouds in the feature matching process, because the descriptiveness of a feature descriptor can directly affect registration precision. Therefore, reasonable parameters are required by the LFSH feature descriptor. In our LFSH feature descriptor, there are four key parameters: the bin sizes N_1, N_2, and N_3 of the three sub-histograms, and the search radius r. The bin sizes of each sub-histogram determine the storage size and the discriminative power of the descriptor. The value of r controls the neighboring size of a query point. It will affect the computational efficiency because large values of r will bring additional samples for statistics. Additionally, small values of r will make the underlying surface contain less discriminative information, and lead to a less distinctive descriptor. Because the scale of the input 3D model is usually unknown, it is difficult to directly obtain a reasonable order of magnitude for the values of r. Consequently, we adopt the percentage of the model size to assign the value of r. The model size is defined as the length of the diagonal line of the model's bounding box.

Please cite this article as: J. Yang et al., A fast and robust local descriptor for 3D point cloud registration, Information Sciences (2016), http://dx.doi.org/10.1016/j.ins.2016.01.095

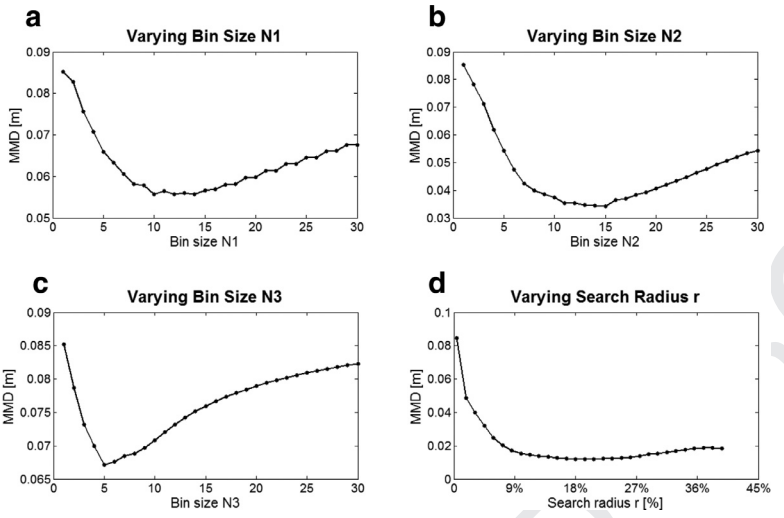

Fig. 3. Effectiveness of the four different parameters (N_1, N_2, N_3, and r) for LFSH with MMD.

Table 1
Parameter settings in Fig. 3.

	Bin size N_1	Bin size N_2	Bin size N_3	Search radius r (%)
Fig. 3(a)	1–30	1	1	13
Fig. 3(b)	1	1–30	1	13
Fig. 3(c)	1	1	1–30	13
Fig. 3(d)	10	15	5	(0.4 – 39)

To analyze the four parameters quantitatively, we introduce a metric named mean match distance (MMD) to evaluate the discriminability of a descriptor, as inspired by Johnson and Hebert [27]. The definition is given as follows:

$$MMD = \frac{1}{N}\sum_{i=1}^{N}\frac{\sum_{j=1}^{k_i}\|p_i - \tilde{p}_j\|}{k_i}, \tag{5}$$

where $\tilde{p}_j$ represents a point in P whose LFSH feature descriptor is similar to p_i, and k_i is the number of these qualified points. MMD describes the correctness of the LFSH matches. More precisely, the higher the number of correct matches, the lower the MMD will be. In our analysis, we select the Stanford Bunny model as our experimental model (see Fig. 6). Fig. 3 shows the impact of the four parameters on the LFSH descriptor. It is obtained by calculating the MMD of the Bunny model with varying bin sizes and search radius repeatedly. Table 1 shows the parameter settings. The step size of the search radius is set to 1.3%, and the sample interval for the three bin sizes is set to 1.

From Fig. 3, one can see that the descriptiveness of LFSH is closely yet nonlinearly related to the four parameters. The available value ranges for bin sizes N_1, N_2, and N_3 can be configured as [8, 15], [10, 17], and [4, 7], respectively. For instance, we set $N_1 = 10$, $N_2 = 15$, and $N_3 = 5$, forming a 30-bin LFSH descriptor; the effect of search radius r can be observed in Fig. 3(d). It is clear that when the search radius is between 10% and 30% of the model size, the values of MMD are rarely changed. Considering the computational efficiency, a scale of approximately 13% of the model size is suitable for LFSH extraction.

Note that, compared with the results using single features shown in Fig. 3(a)–(c), the MMD decreased considerably when the three features were combined with a proper search radius, as shown Fig. 3(d). The fusion of faint correlated features endows the feature histogram with far greater discriminative power.

4. Coarse-to-fine 3D registration algorithm

In practice, a state-of-the-art point clouds registration algorithm should be able to automatically and accurately align two point clouds under a variety of nuisances, including small overlaps, noise, and varying point cloud resolutions. The goal of 3D registration is to find the rotation matrix $\mathbf{R}$ and the translation vector $\mathbf{t}$ that can minimize the distance between two point clouds, i.e., the source point cloud P_s and the target point cloud P_T, that is:

$$\min \sum \|\mathbf{R}\cdot P_S + \mathbf{t} - P_T\|. \tag{6}$$

Please cite this article as: J. Yang et al., A fast and robust local descriptor for 3D point cloud registration, Information Sciences (2016), http://dx.doi.org/10.1016/j.ins.2016.01.095

In this section, we propose a coarse-to-fine point cloud registration algorithm based on the LFSH descriptor. It consists of four major processes: feature extraction, correspondence generation, OSAC-based outlier elimination, and the refined registration.

4.1. LFSH feature extraction

Given a source point cloud P_S and a target point cloud P_T, the first process is to extract LFSH features for the two point clouds. In most cases, the raw input datasets are extremely large for matching algorithms, and would cause high computation costs. In general, there are two speed-up strategies. One is to select a set of feature points from the raw datasets, e.g., THRIFT [15] and MeshHOG [55]. The other is to randomly sample points from the source data and then match these sample points against all the points in the target data [26]. Although both methods significantly reduce storage and time costs, they always suffer from the low accuracy of the registration results, owing to the nonuniform distribution and the instability of the selected points. Here, we use the real-time compression algorithm proposed in [28] to uniformly downsample P_S and P_T, resulting in two low-resolution models $P_{\widetilde{S}} = \{p^i_{\widetilde{S}} | i = 1, 2, \ldots, N_{\widetilde{S}}\}$ and $P_{\widetilde{T}} = \{p^i_{\widetilde{T}} | i = 1, 2, \ldots, N_{\widetilde{T}}\}$. Compared with the above two classes of acceleration methods, our strategy ensures alignment that is more consistent, because the essential shape geometry of input datasets is well preserved. Then, we compute the LFSH feature descriptors for each point in $P_{\widetilde{S}}$ and $P_{\widetilde{T}}$, which are denoted as $F_{\widetilde{S}} = \{f^i_{\widetilde{S}} | i = 1, 2, \ldots, N_{\widetilde{S}}\}$ and $F_{\widetilde{T}} = \{f^i_{\widetilde{T}} | i = 1, 2, \ldots, N_{\widetilde{T}}\}$.

4.2. Correspondence generation

Because a rigid transformation can be specified by three or more correct point-to-point correspondences [6], the registration problem becomes easy if the correspondences between the source and target data are perfectly established. Under circumstances in which no artificial markers are used, the essential task is to estimate reasonable correspondences by matching the extracted LFSH feature sets $F_{\widetilde{S}}$ and $F_{\widetilde{T}}$.

For a feature $f^i_{\widetilde{S}}$ from $F_{\widetilde{S}}$, we find its k-nearest features $\{f^{i1}_{\widetilde{T}}, f^{i2}_{\widetilde{T}}, \ldots, f^{ik}_{\widetilde{T}}\}$ from $F_{\widetilde{T}}$. Their associated points $\{p^{i1}_{\widetilde{T}}, p^{i2}_{\widetilde{T}}, \ldots, p^{ik}_{\widetilde{T}}\}$ are considered as the k correspondence candidates for $p^i_{\widetilde{S}}$, which are denoted by $e_i = \{(p^i_{\widetilde{S}}, p^{ij}_{\widetilde{T}}) | j = 1, 2, \ldots, k\}$. In particular, the KD-tree algorithm is employed for efficient search in the LFSH feature space, and the popular distance metric L2 norm is used for comparing LFSH descriptors. Once the correspondence candidates for all points in $P_{\widetilde{S}}$ have been determined, we finally obtain a correspondence candidate set $E = \{e_i | i = 1, 2, \ldots, N_S\}$ for $P_{\widetilde{S}}$ and $P_{\widetilde{T}}$. Note that E contains $k \cdot N_S$ correspondences in total.

4.3. OSAC-based outliers elimination

Thus far, we have created a correspondence candidate set E for $P_{\widetilde{S}}$ and $P_{\widetilde{T}}$. Because the input datasets are usually partially overlapped, not every point in $P_{\widetilde{S}}$ is supposed to have a corresponding point in $P_{\widetilde{T}}$. Therefore, the next task is to remove the outliers from E.

In this paper, we adopt a two-stage outliers elimination strategy. Because the L2 distance between the correct matched points should be equal to 0 ideally, we employ a threshold d_f to remove the majority of the outliers in the first step. d_f is decided by computing the mean μ_f and standard deviation σ_f of the Euclidean distances between the LFSH descriptors of every point pair in set E. It is calculated as

$$d_f = \mu_f - \alpha \cdot \sigma_f, \tag{7}$$

where the parameter α is used to control the size of the output correspondence set. In fact, the size of α has a slight impact on the precision of the registration result, because we have prepared a robust outliers removal method at the second stage to refine the result. Nevertheless, proper α values are beneficial for the time efficiency of our registration algorithm, and we recommend setting it between 0.5 and 1.5. As a result, the correspondence candidates in E that have larger distances between their LFSH feature descriptors than d_f are filtered. It returns a coarse correspondence set $C = \{(c^i_{\widetilde{S}}, c^i_{\widetilde{T}}) | i = 1, 2, \ldots, m, c^i_{\widetilde{S}} \in P_{\widetilde{S}}, c^i_{\widetilde{T}} \in P_{\widetilde{T}}\}$, which consists of m correspondences.

Because symmetry in an object may cause two or more points having similar LFSH features, the resultant erroneous correspondences can be difficult to distinguish using only distance constrains. Consequently, we introduce an optimized sample consensus (OSAC) algorithm to remove the outliers further. This is necessary because the traditional random sample consensus (RANSAC) [12] method has difficulty recognizing the correct correspondences when the input datasets are highly self-similar. In the OSAC algorithm, we introduce an error metric

$$\mathcal{D}_{\text{avg}}(P_1, P_2) = \begin{cases} \dfrac{1}{\widehat{N}} \sum\limits_{i=1}^{\widehat{N}} d(\widehat{p}^i_1, P_2) & \text{if } \dfrac{\widehat{N}}{\min\{N_1, N_2\}} > \delta \\ \infty & \text{otherwise} \end{cases}, \tag{8}$$

where $P_1 = \{p^i_1 | i = 1, 2, \ldots, N_1\}$ and $P_2 = \{p^i_2 | i = 1, 2, \ldots, N_2\}$ represent two given point clouds, $\widehat{p}^i_1$ is a point in P_1 that satisfies its distance from P_2 being smaller than a distance threshold d_τ, $\widehat{N}$ is the number of qualified points, and δ is a parameter

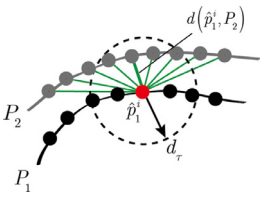

Fig. 4. Illustration of the key parameters in $\mathcal{D}_{\text{avg}}(P_1, P_2)$. The red point is an example of $\hat{p}_1^i$, the green lines represent the distances from $\hat{p}_1^i$ to the points in P_2, where the bold green line denotes the point-to-surface distance from $\hat{p}_1^i$ to P_2. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

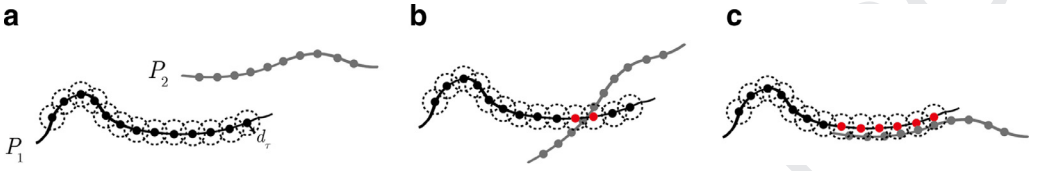

Fig. 5. Three typical spatial relationships between two point clouds in 2D. (a) Disjointed. (b) Intersected. (c) Spatially close. Red points denote points in P_1 that satisfy their point-to-surface distances to P_2 being smaller than d_τ. The $\mathcal{D}_{\text{avg}}(P_1, P_2)$ only takes effect in case (c). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

used to judge if P_1 and P_2 are spatially close. Fig. 4 shows the illustration of the key parameters in $\mathcal{D}_{\text{avg}}(P_1, P_2)$. The red point is an example of $\hat{p}_1^i$, and the distance between $\hat{p}_1^i$ and P_2 is defined as the point-to-surface distance in [39]. It is given by

$$d(\hat{p}_1^i, P_2) = \min_{j=1,2,\ldots,N_2} \|\hat{p}_1^i - p_2^j\|. \tag{9}$$

In Fig. 4, the bold green line denotes the point-to-surface distance of $\hat{p}_1^i$ as it ranks the smallest among the distances from $\hat{p}_1^i$ to the points in P_2; the red point is a qualified point in P_1 because its point-to-surface distance is smaller than d_τ. The values of d_τ determine how we define whether a point in P_1 is close to P_2. Ideally, $d(\hat{p}_1^i, P_2)$ would be close to zero when P_1 and P_2 have overlapping regions and are precisely registered; thus, small values of d_τ are proper. Because input datasets are usually partially overlapped, the range of δ is (0, 1). Empirically, we set $\delta = 0.3$ in our implementation.

The purpose of defining $\mathcal{D}_{\text{avg}}(P_1, P_2)$ is to quantitatively compute the distance between two point clouds when they are close to each other. It is based on the assumption that the distance between two well-aligned point clouds equals zero. Fig. 5 shows three typical spatial relationships between two point clouds, and $\mathcal{D}_{\text{avg}}(P_1, P_2)$ returns an effective value only in the case of Fig. 5(c). A summary of the stages for the implementation of OSAC is presented in Algorithm 1.

Algorithm 1 OSAC Algorithm.

Require: the coarse correspondence set C
Ensure: the optimum transformation $\widehat{M}$
1: Initial $i = 1$ and the maximum iteration times N_{iter}.
2: **while** $i < N_{iter}$ **do**
3: Select x ($x \geq 3$) samples from C. Make sure the pairwise distances of $\{c_{\widetilde{S}}^1, c_{\widetilde{S}}^2, \ldots, c_{\widetilde{S}}^x\}$ and $\{c_{\widetilde{T}}^1, c_{\widetilde{T}}^2, \ldots, c_{\widetilde{T}}^x\}$ are larger than $d_{\min}$.
4: Compute the determined rigid transformation M_i and transform $P_{\widetilde{S}}$ to $P'_{\widetilde{S}_i}$.
5: Compute $\mathcal{D}_{\text{avg}}(P'_{\widetilde{S}_i}, P_{\widetilde{T}})$ using Eq. 8.
6: Push $\mathcal{D}_{\text{avg}}(P'_{\widetilde{S}_i}, P_{\widetilde{T}})$ into a stack $\{\mathcal{D}_{\text{avg}}(P'_{\widetilde{S}_1}, P_{\widetilde{T}}), \mathcal{D}_{\text{avg}}(P'_{\widetilde{S}_2}, P_{\widetilde{T}}), \ldots, \mathcal{D}_{\text{avg}}(P'_{\widetilde{S}_i}, P_{\widetilde{T}})\}$.
7: **if** $\mathcal{D}_{\text{avg}}(P'_{\widetilde{S}_i}, P_{\widetilde{T}}) < \mathcal{D}_\tau$ **do**
8: Break.
9: **end if**
10: $i = i + 1$.
11: **end while**
12: Let $\widehat{M} = \arg\min \{\mathcal{D}_{\text{avg}}(P'_{\widetilde{S}_1}, P_{\widetilde{T}}), \mathcal{D}_{\text{avg}}(P'_{\widetilde{S}_2}, P_{\widetilde{T}}), \ldots, \mathcal{D}_{\text{avg}}(P'_{\widetilde{S}_i}, P_{\widetilde{T}})\}$.
13: return $\widehat{M}$.

In Algorithm 1, $d_{\min}$ is a user-defined minimum distance. It is employed as a rule for the selection of correspondence samples suggested in [44]. The maximum iteration count N_{iter} and the error metric threshold Δ_τ are two stop conditions for the iterations. At each iteration, we first measure the correctness of the current transformation by computing $\mathcal{D}_{\text{avg}}(P'_{\widetilde{S}_i}, P_{\widetilde{T}})$,

Please cite this article as: J. Yang et al., A fast and robust local descriptor for 3D point cloud registration, Information Sciences (2016), http://dx.doi.org/10.1016/j.ins.2016.01.095

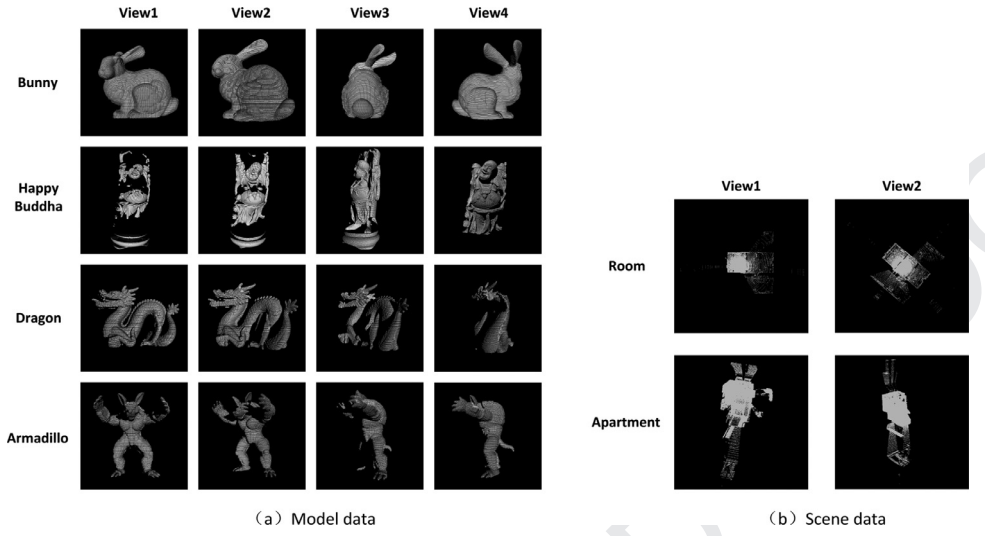

Fig. 6. Experimental dataset.

and then choose the transformation that yielded the minimum error metric to roughly align P_S to P_T. Compared with RANSAC, OSAC judges the correctness of the transformation from a global perspective instead of only taking the coarse correspondence set into consideration.

4.4. Registration refinement

Having roughly registered two point clouds, we are left with the task of further minimizing the registration error between them. This is accomplished by executing the Levenberg–Marquardt iterative closest point (LM-ICP) algorithm [13], where a non-linear local optimization method is used to minimize the residual error. This process brings two point clouds closer together, and creates a seamless shape.

5. Experimental results and discussion

In this section, we extensively examine the performance of the proposed LFSH descriptor, OSAC algorithm, and the coarse-to-fine 3D registration algorithm. The experimental dataset (Fig. 6) contains four models ("Bunny," "Happy Buddha," "Dragon," and "Armadillo") obtained from the Stanford 3D Scanning Repository[1], and two indoor scenes ("Room"[2] and "Apartment"[3]) obtained from the Point Cloud Library (PCL) website [46] and the dataset provided in [41], respectively.

In our implementation, we set bin sizes $N_1 = 10, N_2 = 15, N_3 = 5$, search radius $r = 13\%$, $\alpha = 1$, $d_\tau = 1\%$, and $\delta = 0.3$ for all experiments described in this paper. Note that we use the percentage of the larger of the two input models for the assignment of r and d_τ; this helps us handle the problem of model scale ambiguity. The parameter settings used in this paper are common for all studied areas. Our algorithm is implemented in Visual C++, OpenGL, and PCL, while all the experiments are conducted on a 3.5 GHz Intel(R) Core 3 processor with 8 GB RAM.

5.1. Performance evaluation of LFSH descriptor

In this experiment, we first examine the performance of feature matching using the proposed LFSH descriptor on the experimental dataset. We then further compare the LFSH descriptor with five existing descriptors.

The percentage of correct correspondences [24] is used as the criteria for quantitative analysis. For each neighboring view pair in Fig. 6, we build a coarse correspondence set according to the method described in Section 4.3. In order to judge the correctness of a correspondence $C^i = (c_S^i, c_T^i)$ in the coarse correspondence set C, we compare its transformation error ϵ_i with a threshold d_ϵ. The transformation error is given by

$$\epsilon_i = \|\mathbf{R}_{GT} \cdot c_S^i + \mathbf{t}_{GT} - c_T^i\|, \tag{10}$$

where $\mathbf{R}_{GT}$ and $\mathbf{t}_{GT}$ represent the ground truth rotation and translation, respectively. $\mathbf{R}_{GT}$ and $\mathbf{t}_{GT}$ are obtained by first manually aligning two point clouds, and then further aligning them using the LM-ICP algorithm. Owing to the differences among

[1] http://graphics.stanford.edu/data/3Dscanrep.
[2] http://pointclouds.org/.
[3] http://projects.asl.ethz.ch/datasets/doku.php.

Please cite this article as: J. Yang et al., A fast and robust local descriptor for 3D point cloud registration, Information Sciences (2016), http://dx.doi.org/10.1016/j.ins.2016.01.095

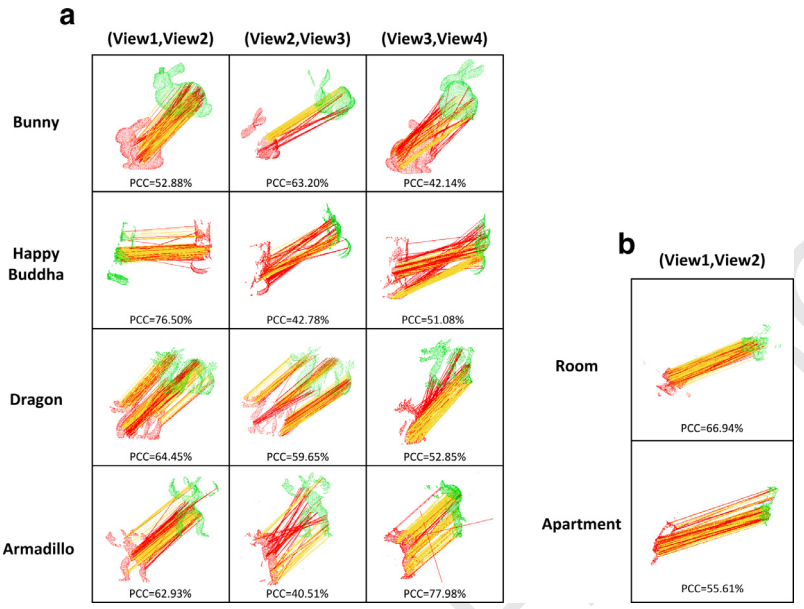

Fig. 7. Feature matching results of the LFSH descriptor on the experimental datasets. (a) Results of model data. (b) Results of scene data. PCC denotes the percentages of the correct correspondences. Red and yellow lines represent erroneous and correct correspondences, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 2
Parameter settings for six feature descriptors.

	Search radius (%)	Support angle (degree)	Dimensionality	Length
Spin image	–	60	15 × 15	225
PFH	13	–	5 × 5 × 5	125
FPFH	13	–	3 × 11	33
SHOT	13	–	8 × 2 × 2 × 10	320
RoPS	13	–	3 × 3 × 3 × 5	135
LFSH	13	–	10 + 15 + 5	30

the scales of experimental data, we set $d_\epsilon = 4$ mm, 6 mm, 4 mm, 6 mm, 0.7 m, and 0.5 m for the Bunny, Happy Buddha, Dragon, Armadillo, Room, and Apartment data, respectively. The feature matching results are shown in Fig. 7. Two observations can be made from the results. First, our LFSH descriptor can achieve high percentages of correct correspondences on most view pairs. Second, for some challenging view pairs that either have low overlapping ratios (e.g., view3 and view4 of Bunny) or contain many scanning holes due to self-occlusion (e.g., view2 and view3 of Happy Buddha), the LFSH descriptor is still able to obtain sufficient correct correspondences. Note that the majority of the contaminated incorrect matches can be further removed using the proposed OSAC algorithm (as demonstrated in Section 5.2).

We then rigorously examine the descriptiveness, robustness, and time efficiency of the LFSH descriptor by comparing it with five state-of-the-art descriptors, including the spin image [26], PFH [45], FPFH [44], SHOT [51], and RoPS [23]. Among these descriptors, the spin image is one of the most cited descriptors in 3D surface matching [23]. FPFH and SHOT have been proved to be quite computationally efficient [2], and RoPS is one of the latest descriptors that outperforms most descriptors in terms of feature matching [20]. The parameter settings for these feature descriptors are listed in Table 2. Note that the neighboring size of the spin image is determined by the support angle, and that the other five feature descriptors are determined by the search radius.

5.1.1. Descriptiveness

Table 3 shows the feature matching results of the six descriptors on Bunny (view1 and view2), Armadillo (view1 and view2), Room, and Apartment data. For each set of data, we generate the coarse correspondence sets using these descriptors according to the method described in Section 4.3. In order to compare fairly, the parameter α is fixed to 1 for all descriptors. Here, we set $d_\epsilon = 3$ mm, 5 mm, 0.5 m, and 0.3 m for the Bunny, Armadillo, Room, and Apartment data, respectively.

From Table 3, we find that the RoPS descriptor achieves the highest percentage of correct correspondences on Bunny and Armadillo data, followed by the proposed LFSH descriptor. The spin image descriptor is inferior to other descriptors on both sets of data. A similar conclusion has been confirmed in [23]. In addition, the performance of PFH and SHOT descriptors fluctuates on two model data sets. Room and Apartment data contain poorer geometric information compared with the

Table 3
Feature matching results of the six descriptors on Bunny, Armadillo, Room, and Apartment data. (*NC* denotes the number of correspondences. *NCC* represents the number of correct correspondences. *PCC* is the percentage of correct correspondences.)

		Spin image	PFH	FPFH	SHOT	RoPS	LFSH
Bunny	*NC*	546	312	247	461	357	234
	NCC	120	82	58	105	168	83
	PCC (%)	21.98	26.28	23.48	22.78	**47.06**	35.5
Armadillo	*NC*	267	319	208	405	404	285
	NCC	82	114	63	180	227	137
	PCC (%)	30.71	35.74	30.29	44.44	**56.19**	48.07
Room	*NC*	449	229	293	464	335	254
	NCC	69	87	90	205	117	135
	PCC (%)	15.37	37.99	30.72	44.18	34.93	**53.15**
Apartment	*NC*	815	1021	1196	1172	892	937
	NCC	68	282	295	306	128	345
	PCC (%)	8.34	27.62	24.67	26.11	14.35	**36.82**

Bunny and Armadillo data. It is remarkable that the Room and Apartment data contain poorer geometric features compared with the model data, and they are therefore more challenging for feature matching. We note that features that mainly encode normal characteristics, e.g., PFH and FPFH descriptors, return relatively low matching accuracy. The spin image and RoPS descriptors achieve worse results for scene data than model data, owing to the ambiguity of LRF caused by plane geometry. The reasons for the superiority of our LFSH descriptor on scene data are twofold. One is that LFSH not only encodes normal characteristics, but also includes other discriminative features such as point density. The other reason is that our LFSH descriptor is not LFR-based; instead, a more robust description is achieved on data with poor geometric information.

5.1.2. Robustness

In this section, we assess the robustness of the six descriptors with respect to Gaussian noise and varying point cloud resolutions. View1 of Bunny is viewed as the target data. First, we add Gaussian noise with different standard deviations for the target data, where the standard deviation of noise is increased from 0 to 5 mm in incremental steps of 0.5 mm. Then, we simplify the target data to varying point cloud resolutions. The number of points in a simplified point cloud is between 10% and 90% of its original value at an interval of 10%. The point count of original view1 of Bunny is 40,256. Finally, 11 copies of noisy source data and 10 copies of simplified source data are generated. Notably, the simplification step in Section 4.1 is neglected for the generation of coarse correspondence sets. The feature matching results with respect to Gaussian noise and varying point cloud resolutions are shown in Fig. 8.

In Fig. 8(a), all the descriptors can achieve a relatively high percentage of correct correspondences under low-level Gaussian noise (the standard deviation is no more than 2 mm). However, the performance of PFH and FPFH descriptors drops quickly when the standard deviation of noise is larger than 2 mm. This is because large-scale noise has a significant influence on the calculation of normals. The SHOT descriptor provides the best performance under high-level noise, followed by the LFSH, spin image, and RoPS descriptors. Fig. 8(b) shows that the RoPS descriptor provides the most stability under high point cloud resolution variations, followed by our LFSH descriptor. The results demonstrate the advantage of using a set of statistics of local features to form a descriptor [23] (as in the case of RoPS and LFSH descriptors). Another observation is that the performance of the six descriptors clearly starts to deteriorate when the simplified data has less than 80% of its original number of points.

Fig. 8 also shows the results of matching the noisy and simplified view1 of Bunny with the original data using our LFSH descriptor. Note that the noisy data is the one that is added with a noise standard deviation of 5 mm, and the simplified data is the one that has 10% of its original number of points. In such challenging cases, the results show that our LFSH descriptor is still workable.

5.1.3. Time efficiency

Fig. 9 shows the feature extraction time of the six descriptors under different levels of point cloud resolution. This experiment aims at examining the time efficiency of feature descriptors for point clouds with different point densities. The experimental data are obtained by simplifying the original bunny model to twelve different point cloud resolutions. The original bunny model contains 362,272 points. For each copy, the feature extraction procedure is performed directly on the given data. Specifically, the feature descriptors are computed for each point in the given point cloud. The time costs of each feature descriptor on the twelve sets of data are then recorded. The entire process is repeated 10 times, and the statistics of the average time are taken as the final results.

In Fig. 9, one can see that our LFSH descriptor is superior to other descriptors in time efficiency at all levels of point cloud resolution, followed by the SHOT, FPFH, spin image, RoPS, and PFH descriptors. In addition, both FPFH and SHOT descriptors are faster than the spin image and RoPS descriptors by approximately an order of magnitude. Because of its high

Please cite this article as: J. Yang et al., A fast and robust local descriptor for 3D point cloud registration, Information Sciences (2016), http://dx.doi.org/10.1016/j.ins.2016.01.095

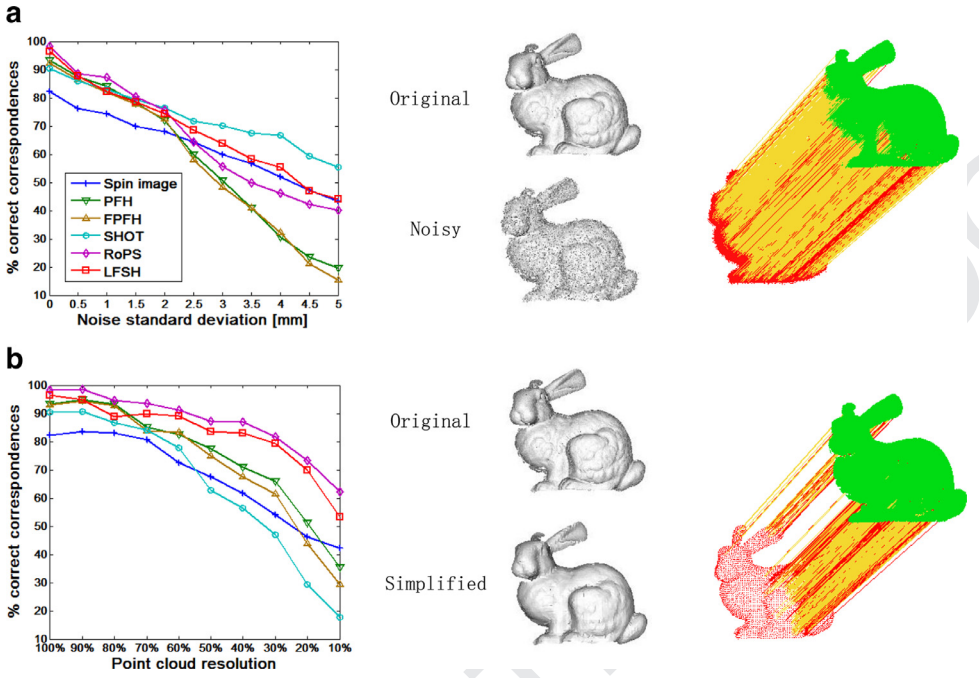

Fig. 8. Robustness evaluation of feature descriptors. (a) Feature matching results with respect to different levels of Gaussian noise, and an example with noise standard deviation of 5 mm. (b) Feature matching results with respect to varying point cloud resolutions, and an example with 10% point cloud resolution. The source point clouds (including the noisy and simplified data) are shown in red, and the target point cloud (the original data) is shown in green. Red and yellow lines indicate erroneous and correct correspondences, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

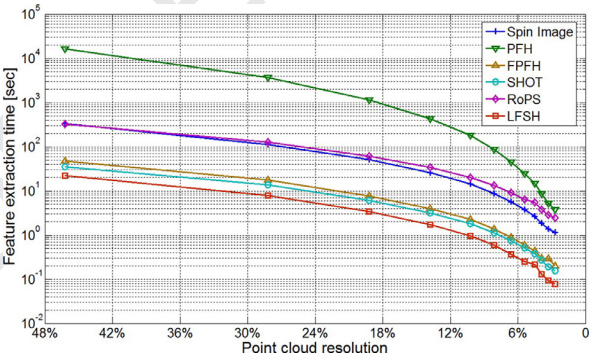

Fig. 9. Time efficiency evaluation of feature descriptors under different levels of point cloud resolution. The *Y*-axis is the logarithmic axis to provide the best view.

computational complexity [44], the PFH descriptor is significantly slower than other descriptors, especially for high point cloud resolution data. In particular, our LFSH descriptor is over two times faster than the FPFH and SHOT descriptors. It is also worth noting that the proposed LFSH descriptor has the lowest dimensionality among the six descriptors, which is beneficial for efficient data storage as well as feature matching.

5.2. Performance of OSAC

In this section, we validate the performance of OSAC on the Armadillo (view3 and view4), Dragon (view1 and view2), and Bunny (view1 and view3) models. Additionally, we compare the OSAC algorithm with the RANSAC algorithm. For each set of data, a coarse correspondence set C is generated. We use both methods to identify an inlier set for C under 1000 iterations. The percentage of correct correspondences of the inlier set is employed as the evaluation criterion. The results of the two methods with different d_ϵ are shown in Fig. 10.

Fig. 10 shows that the OSAC algorithm performs slightly better than RANSAC on the Armadillo and Dragon data, and outperforms RANSAC by a large margin under all threshold levels for Bunny data. For further visual assessment, the results with $d_\epsilon = 4$ mm are shown in Fig. 11. For the Armadillo data, the coarse correspondence set contains a certain amount of

Please cite this article as: J. Yang et al., A fast and robust local descriptor for 3D point cloud registration, Information Sciences (2016), http://dx.doi.org/10.1016/j.ins.2016.01.095

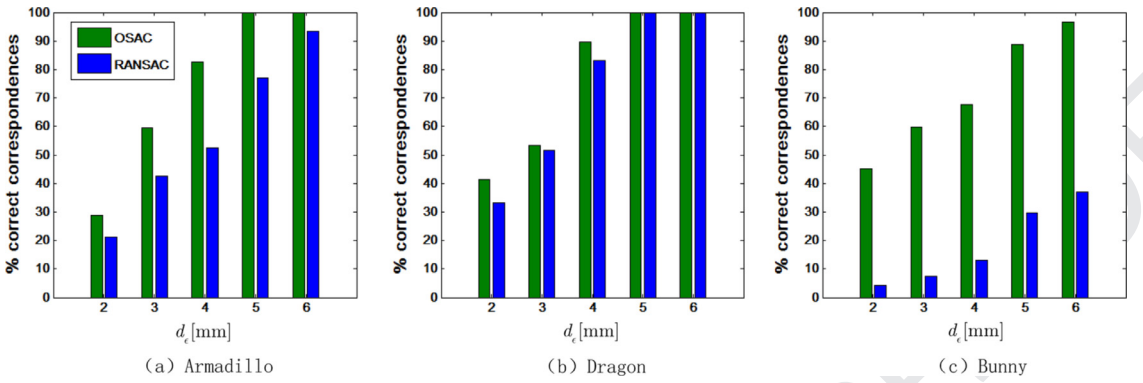

Fig. 10. Percentage of identified correct correspondences of RANSAC and OSAC with different d_ϵ.

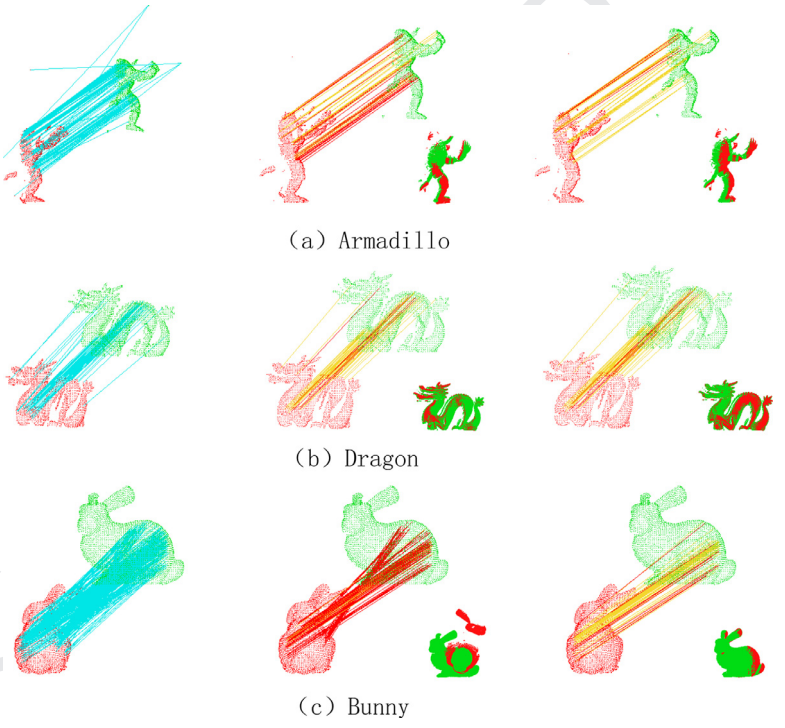

Fig. 11. Comparison between RANSAC and OSAC with $d_\epsilon = 4$ mm. From left to right: the coarse correspondence sets, the results of RANSAC, and the results of OSAC. Cyan, red, and yellow lines represent correspondences in the coarse correspondence set, erroneous correspondences, and correct correspondences, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

inconsistent correspondences, which are caused by the outliers inside the raw scanned data; both RANSAC and OSAC return a quite consistent inlier set. Under the current threshold, OSAC performs better than RANSAC. For the Dragon data, the coarse correspondence set appears to be of high quality; thus, both OSAC and RANSAC achieve high percentages of correct correspondences. The rigid transformations estimated by both methods can successfully bring the source model fairly close to the target model. For the Bunny data, the coarse correspondence set of this model is quite challenging for grouping. This is because the overlapping ratio between view1 and view3 is quite low, and the view3 of Bunny (shown in red in Fig. 11(c)) is highly self-similar. Consequently, the resultant correspondences are fairly inconsistent. In this case, the inlier correspondences returned by RANSAC are still severely contaminated by incorrect matches, which leads to the failure of coarse registration. In contrast, the OSAC algorithm can still identify sufficient correct correspondences, and successfully estimate a reasonable transformation for the registration.

Please cite this article as: J. Yang et al., A fast and robust local descriptor for 3D point cloud registration, Information Sciences (2016), http://dx.doi.org/10.1016/j.ins.2016.01.095

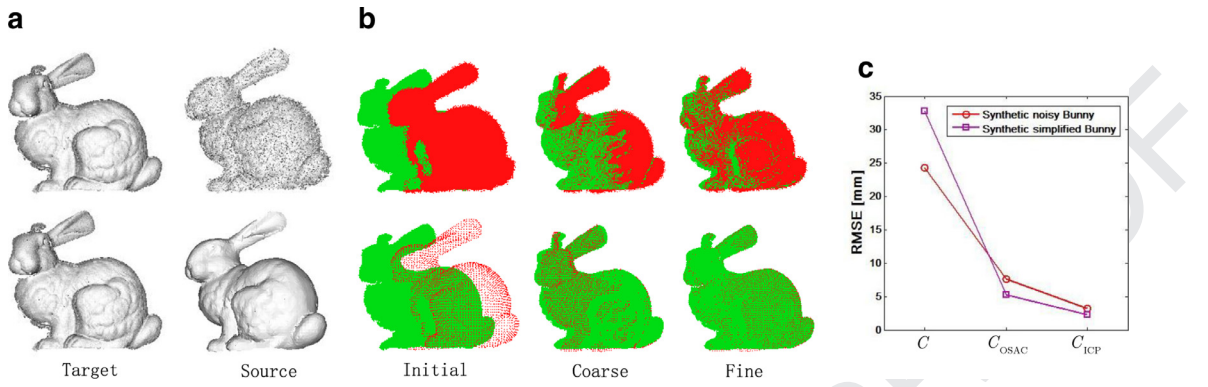

Fig. 12. Registration results of synthetic data. (a) The noisy and simplified source data (view2 of Bunny), and the original target data (view1 of Bunny). (b) From left to right: initial positions of the input data, coarse registration, and fine registration. (c) RMSE of the registration.

5.3. Validation of the registration algorithm

To verify the effectiveness and robustness of the registration algorithm, three types of data are used in this section, including synthetic challenging data, real-world scanned model data, and scene data. The registration results are first evaluated by visualization, and then quantitatively assessed using the root mean square error (RMSE) [43] between corresponding points. The RMSE is defined as

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N_C} \left\| \mathbf{R}_{GT} \cdot c_S^i + \mathbf{t}_{GT} - c_T^i \right\|^2}{N_C}}, \tag{11}$$

where c_S^i and c_T^i are two corresponding points, and N_C is the size of the correspondence set. Specifically, the RMSE values of the coarse correspondence set C, the identified inlier set C_{OSAC} by OSAC, and the correspondence set C_{ICP} returned by LM-ICP are presented. The smaller the RMSE, the better the two point clouds are aligned.

5.3.1. Synthetic data

In this experiment, the synthetic data, which are generated by adding Gaussian noise and changing point cloud resolution, are used to validate our registration algorithm. View1 of Bunny is selected as the target data. The noisy and simplified data obtained by view2 of Bunny are taken as the source data, respectively. The point count of the simplified data is 10% of its original value, and the standard deviation of Gaussian noise is 5 mm. To preserve these synthetic nuisances, the simplification step before feature extraction is neglected and the LFSH descriptors for each point in the input data are computed. Fig. 12 shows the registration results.

Fig. 12 shows that our algorithm successfully brings the noisy and simplified source data close to the target data. In addition, the RMSE value presents a decreasing trend under the registration course, which demonstrates that our algorithm is quite robust to noise and point cloud resolution variation.

5.3.2. Model data

Fig. 13 shows the registration results of model data (view1 and view4 of Happy Buddha, view1 and view2 of Armadillo, view1 and view3 of Dragon). The input scans are partially overlapped, and given in arbitrary initial positions. It shows that the input scans have been brought into close alignment by our algorithm, and the RMSE values decline significantly after the registration.

5.3.3. Indoor scene data

Fig. 14 shows the registration results of indoor scene data. Compared to model data, indoor scene data usually contain poorer geometric features. Moreover, the raw indoor scene data contain a certain number of outliers. We can see that accurate registration for scene data is accomplished, and two scans are well converged. Moreover, the RMSE values are low after the registration.

5.4. Registration timing conclusions

Timing statistics of the registration results (shown in Figs. 13 and 14) are listed in Table 4. In general, the whole registration courses for all experimental data require only a few seconds. In particular, the feature extraction time of our algorithm for all data takes no more than one second. On one hand, it benefits from the adopted simplification strategy; on the other, it benefits from the high computational efficiency of the LFSH descriptor. The main time consumption of our registration algorithm lies in the feature matching and refining processes, where iterations are needed. However, we find that the LM-ICP

Please cite this article as: J. Yang et al., A fast and robust local descriptor for 3D point cloud registration, Information Sciences (2016), http://dx.doi.org/10.1016/j.ins.2016.01.095

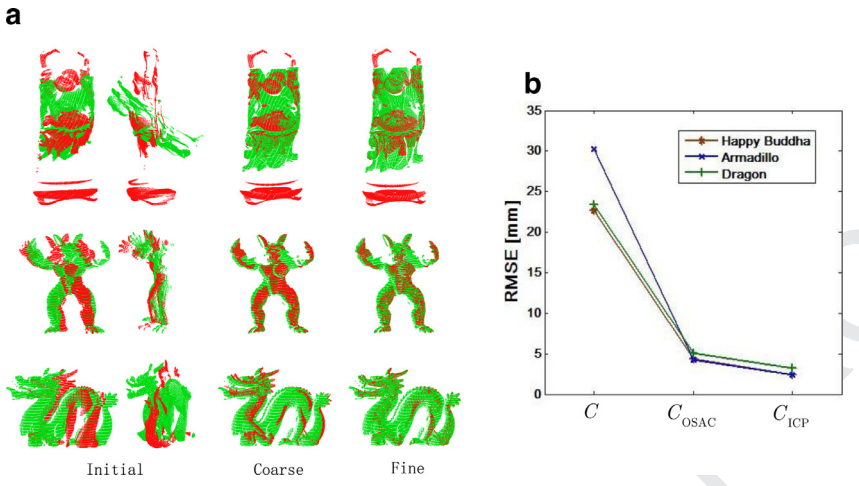

Fig. 13. Registration results of model data. (a) From left to right: two views of the initial positions of the input data, coarse registration, and fine registration. (b) RMSE of the registration.

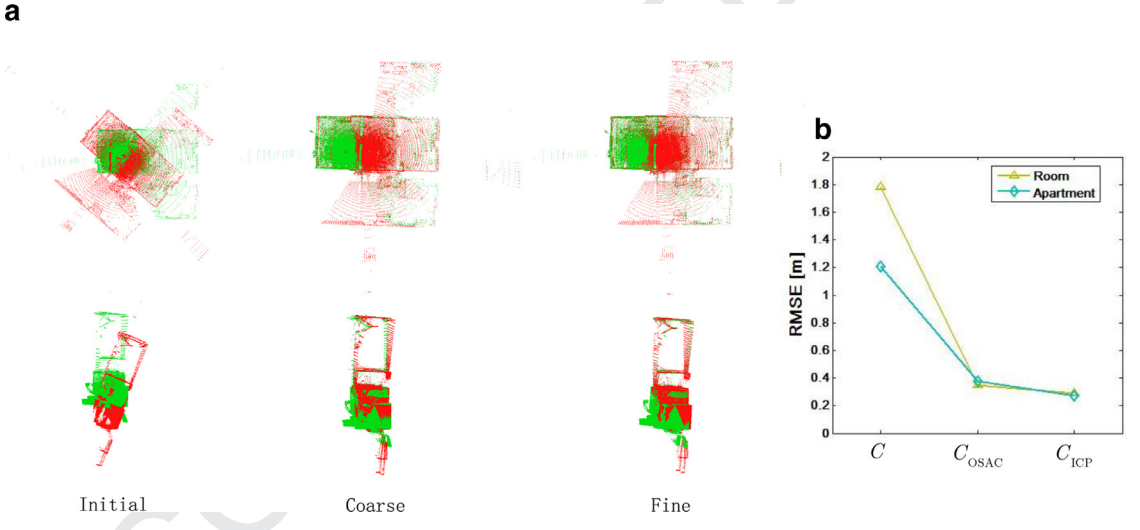

Fig. 14. Registration results of scene data. (a) From left to right: initial positions of the input data, coarse registration, and fine registration. (b) RMSE of the registration.

Table 4
Registration timing statistics. (T_{FE}, T_{FM}, T_{LM-ICP} and T_{sum} represent the feature extraction time, feature matching time, LM-ICP refining time, and total registration time, respectively.)

Data	Source size	Target size	T_{FE}(s)	T_{FM}(s)	*Iterations*	$T_{LM\text{-}ICP}$ (s)	T_{sum} (s)
Happy Buddha	57,499	59,544	**0.624**	4.211	3	0.813	5.648
Armadillo	22,495	26,769	**0.266**	2.586	4	1.256	4.108
Dragon	22,092	41,841	**0.717**	2.303	3	0.546	3.566
Room	112,624	112,586	**0.398**	2.610	2	3.213	6.221
Apartment	369,977	370,277	**0.737**	1.104	3	3.893	5.734

algorithm converges quickly for all experimental data. This is because our algorithm promises highly precise coarse registration. It helps us achieve efficient registration refining. Note that the time consumed by LM-ICP for Room and Apartment data is higher than that for the other sets of data, owing to data redundancy.

6. Conclusion and future work

In this paper, a fast and robust local descriptor was proposed for the problem of correspondence search in 3D point clouds. By combining a set of low-dimensional but distinctive geometric features, LFSH gains a minimal loss on local shape

Please cite this article as: J. Yang et al., A fast and robust local descriptor for 3D point cloud registration, Information Sciences (2016), http://dx.doi.org/10.1016/j.ins.2016.01.095

descriptions and is robust to noise and varying point cloud resolutions. Based on LFSH, a novel 3D point cloud registration algorithm was developed. Using a real-time simplification strategy (rather than computing key points) and grouping the correspondences from a global perspective, the proposed registration method equipped with LFSH was highly efficient and effective for point cloud registration.

The performance of the LFSH descriptor was extensively evaluated in terms of descriptiveness, robustness, and time efficiency. It has been compared with several state-of-the-art descriptors. Our LFSH descriptor outperforms most existing descriptors in both descriptiveness and robustness. Although the LFSH descriptor is inferior to the RoPS descriptor in descriptiveness on model data, and ranks second in the robustness aspect, it is faster than the RoPS descriptor by approximately an order of magnitude, and is several times faster than the FPFH and SHOT descriptors. Note that the proposed registration algorithm based on the LFSH descriptor is fully automatic. It can be performed without human intervention, and no prior knowledge about the initial position of the datasets is required. The registration algorithm is efficient and accurate for two reasons. First, feature extraction is applied on simplified datasets and the LFSH descriptor is quite efficient to compute. Second, the rigid transformation determined by OSAC can successfully bring the source data to the convergence area of LM-ICP, avoiding being trapped in a local minimum. Moreover, our registration algorithm has been validated on a variety of datasets, ranging from synthetic models to challenging real-scene laser scans. The registration results showed that our registration algorithm is very effective, and robust to common nuisances including noise and point cloud resolution variation.

In the future, there are three interesting directions for further research. One is to integrate more spatial information into the feature descriptor, to further strengthen its descriptiveness. In this paper, the local geometric information is encoded from several aspects with respect to an LRA. The employed LRA has two advantages: it is fast to construct and robust to noise. Unfortunately, the LRA also discards some spatial information, because only 1D statistics histograms are encoded. Consequently, a stable LRF would endow the feature descriptor with more discriminative information. Thus, it is meaningful to solve the problem of establishing a unique and robust LRF in the presence of noise and varying point cloud resolutions, and then re-encode the three local geometric features with respect to it. The second focuses on the compactness of the proposed LFSH descriptor. We use a concatenation operation to integrate the three employed sub-features in this paper. It is simple but may cause redundancy. A more reasonable feature fusion technique, such as a multi-view learning based approach, could be considered to obtain a more compact LFSH descriptor. The last one is to enrich the proposed LFSH descriptor with photometric cues. Currently, many 3D sensors are able to acquire the texture of a shape; these include stereo sensors, structure-from-motion systems, as well as the recently proposed *Kinect* device by Microsoft. When the scanned objects exhibit poor geometric features (e.g., planar or spherical surfaces), matching the LFSH descriptors would return numerous erroneous correspondences. In that case, the fusion of both geometric and photometric information would be beneficial.

Acknowledgment

The authors would like to acknowledge the Stanford 3D Scanning Repository, the PCL library, and the Autonomous Systems Lab (ASL) for making their datasets available to us. This work is supported by the National High Technology Research and Development Program of China under Grant 2015AA015904 and the China Postdoctoral Science Foundation under Grant 2014M562028.

References

[1] A. Albarelli, E. Rodolà, A. Torsello, Fast and accurate surface alignment through an isometry-enforcing game, Pattern Recognit. 48 (7) (2015) 2209–2226.
[2] L.A. Alexandre, 3d descriptors for object and category recognition: a comparative evaluation, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)Workshop on Color-Depth Camera Fusion in Robotics, 2012.
[3] K. Alhamzi, M. Elmogy, S. Barakat, 3d object recognition based on local and global features using point cloud library, Int. J. Adv. Comput. Technol. 7 (3) (2015) 43.
[4] J. Assfalg, M. Bertini, A.D. Bimbo, P. Pala, Content-based retrieval of 3-d objects using spin image signatures, IEEE Trans. Multimed. 9 (3) (2007) 589–599.
[5] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, IEEE Trans. Pattern Anal. Mach. Intell. 24 (4) (2002) 509–522.
[6] P.J. Besl, N.D. McKay, Method for registration of 3-d shapes, IEEE Trans. Pattern Anal. Mach. Intell. 14 (2) (1992) 239–256.
[7] O. Carmichael, D. Huber, M. Hebert, Large data sets and confusing scenes in 3-d surface matching and recognition, in: Proceedings of the Second International Conference on 3-D Digital Imaging and Modeling, IEEE, 1999, pp. 358–367.
[8] C.S. Chen, Y.P. Hung, J.B. Cheng, Ransac-based darces: A new approach to fast automatic registration of partially overlapping range images, IEEE Trans. Pattern Anal. Mach. Intell. 21 (11) (1999) 1229–1234.
[9] J.C. Cheng, H.S. Don, A graph matching approach to 3-d point correspondences, Int. J. Pattern Recognit. Artif. Intell. 5 (03) (1991) 399–412.
[10] C.S. Chua, R. Jarvis, Point signatures: A new representation for 3d object recognition, Int. J. Comput. Vis. 25 (1) (1997) 63–85.
[11] H.Q. Dinh, S. Kropac, Multi-resolution spin-images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, IEEE, 2006, pp. 863–870.
[12] 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.
[13] A.W. Fitzgibbon, Robust registration of 2d and 3d point sets, Image Vis. Comput. 21 (13) (2003) 1145–1153.
[14] A. Flint, A. Dick, A. Van den Hengel, Local 3d structure recognition in range images, IET Comput. Vis. 2 (4) (2008) 208–217.
[15] A. Flint, A.R. Dick, A. Van Den Hengel, Thrift: Local 3d structure recognition., in: Proceedings of the Ninth Conference on Digital Image Computing: Techniques and Applications, vol. 7, 2007, pp. 182–188.

Please cite this article as: J. Yang et al., A fast and robust local descriptor for 3D point cloud registration, Information Sciences (2016), http://dx.doi.org/10.1016/j.ins.2016.01.095

[16] A. Frome, D. Huber, R. Kolluri, T. Bülow, J. Malik, Recognizing objects in range data using regional point descriptors, in: Proceedings of the European Conference on Computer Vision, Springer, 2004, pp. 224–237.
[17] N. Gelfand, N.J. Mitra, L.J. Guibas, H. Pottmann, Robust global registration, in: Proceedings of Symposium on Geometry Processing, vol. 2, 2005, p. 5.
[18] M. Gong, Y. Wu, Q. Cai, W. Ma, A. Qin, Z. Wang, L. Jiao, Discrete particle swarm optimization for high-order graph matching, Inf. Sci. 328 (2016) 158–171.
[19] A. Gruen, D. Akca, Least squares 3d surface and curve matching, ISPRS J. Photogramm. Remote Sens. 59 (3) (2005) 151–174.
[20] 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.
[21] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, An integrated framework for 3-d modeling, object detection, and pose estimation from point-clouds, IEEE Trans. Instrum. Meas. 64 (3) (2015) 683–693.
[22] 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. (2015) 1–24.
[23] 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.
[24] 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.
[25] Y. Guo, F. Sohel, M. Bennamoun, J. Wan, M. Lu, A novel local surface feature for 3d object recognition under clutter and occlusion, Inf. Sci. 293 (2015) 196–213.
[26] A.E. Johnson, M. Hebert, Surface matching for object recognition in complex three-dimensional scenes, Image Vis. Comput. 16 (9) (1998) 635–651.
[27] 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.
[28] J. Kammerl, N. Blodow, R.B. Rusu, S. Gedikli, M. Beetz, E. Steinbach, Real-time compression of point cloud streams, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2012, pp. 778–785.
[29] R. Liang, W. Shen, X.-X. Li, H. Wang, Bayesian multi-distribution-based discriminative feature extraction for 3d face recognition, Inf. Sci. (2015).
[30] A. Liu, Z. Wang, W. Nie, Y. Su, Graph-based characteristic view set extraction and matching for 3d model retrieval, Inf. Sci. (2015).
[31] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2) (2004) 91–110.
[32] K. Lu, Q. Wang, J. Xue, W. Pan, 3d model retrieval and classification by semi-supervised learning with content-based similarity, Inf. Sci. 281 (2014) 703–713.
[33] 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.
[34] T. Masuda, Log-polar height maps for multiple range image registration, Comput. Vis. Image Underst. 113 (11) (2009) 1158–1169.
[35] A.S. Mian, M. Bennamoun, R.A. Owens, Automatic correspondence for 3d modeling: an extensive review, Int. J. Shape Model. 11 (02) (2005) 253–291.
[36] A.S. Mian, M. Bennamoun, R.A. Owens, A novel representation and feature matching algorithm for automatic pairwise registration of range images, Int. J. Comput. Vis. 66 (1) (2006) 19–40.
[37] N.J. Mitra, N. Gelfand, H. Pottmann, L. Guibas, Registration of point cloud data from a geometric optimization perspective, in: Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing, ACM, 2004, pp. 22–31.
[38] N.J. Mitra, A. Nguyen, L. Guibas, Estimating surface normals in noisy point cloud data, Int. J. Comput. Geom. Appl. 14 (2004) 261–276.
[39] M. Pauly, M. Gross, L.P. Kobbelt, Efficient simplification of point-sampled surfaces, in: Proceedings of the Conference on Visualization, IEEE, 2002, pp. 163–170.
[40] A. Petrelli, L.D. Stefano, On the repeatability of the local reference frame for partial shape matching, in: Proceedings of the 2011 IEEE International Conference on Computer Vision (ICCV), IEEE, 2011, pp. 2244–2251.
[41] F. Pomerleau, M. Liu, F. Colas, R. Siegwart, Challenging data sets for point cloud registration algorithms, Int. J. Robot. Res. 31 (14) (2012) 1705–1711.
[42] S. Ruiz-Correa, L.G. Shapiro, M. Melia, A new signature-based method for efficient 3-d object recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, IEEE, 2001, pp. I–769.
[43] S. Rusinkiewicz, M. Levoy, Efficient variants of the ICP algorithm, in: Proceedings of the Third International Conference on 3-D Digital Imaging and Modeling, IEEE, 2001, pp. 145–152.
[44] R.B. Rusu, N. Blodow, M. Beetz, Fast point feature histograms (fpfh) for 3d registration, in: Proceedings of IEEE International Conference on Robotics and Automation, IEEE, 2009, pp. 3212–3217.
[45] R.B. Rusu, N. Blodow, Z.C. Marton, M. Beetz, Aligning point cloud views using persistent feature histograms, in: Proceedings of the Twenty First IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2008, pp. 3384–3391.
[46] R.B. Rusu, S. Cousins, 3d is here: Point cloud library (pcl), in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2011, pp. 1–4.
[47] R.B. Rusu, Z.C. Marton, N. Blodow, M. Beetz, Persistent point feature histograms for 3d point clouds, in: Proceedings of the Tenth International Conference on Intelligent Autonomous Systems (IAS-10), 2008, pp. 119–128.
[48] J. Salvi, C. Matabosch, D. Fofi, J. Forest, A review of recent range image registration methods with accuracy evaluation, Image and Vision Computing 25 (5) (2007) 578–596.
[49] F. Stein, G. Medioni, Structural indexing: Efficient 3-d object recognition, IEEE Trans. Pattern Anal. Mach. Intell. 14 (2) (1992) 125–145.
[50] Y. Sun, J. Paik, A. Koschan, D.L. Page, M.A. Abidi, Point fingerprint: a new 3-d object representation scheme, IEEE Trans. Syst. Man Cybern. Part B 33 (4) (2003) 712–717.
[51] F. Tombari, S. Salti, L. Di Stefano, Unique signatures of histograms for local surface description, in: Proceedings of the European Conference on Computer Vision, Springer, 2010, pp. 356–369.
[52] J.V. Wyngaerd, L. Van Gool, R. Kock, M. Proesmans, Invariant-based registration of surface patches, in: Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 1, IEEE, 1999, pp. 301–306.
[53] C. Xu, D. Tao, C. Xu, A survey on multi-view learning, arXiv:1304.5634. (2013).
[54] C. Xu, D. Tao, C. Xu, Multi-view intact space learning, IEEE Trans. Pattern Anal. Mach. Intell. (2015).
[55] A. Zaharescu, E. Boyer, R. Horaud, Keypoints and local descriptors of scalar functions on 2d manifolds, Int. J. Comput. Vis. 100 (1) (2012) 78–98.

Please cite this article as: J. Yang et al., A fast and robust local descriptor for 3D point cloud registration, Information Sciences (2016), http://dx.doi.org/10.1016/j.ins.2016.01.095