Elsevier Editorial System(tm) for Information Sciences
Manuscript Draft

Manuscript Number: INS-D-13-1432R3

Title: A Novel Local Surface Feature for 3D Object Recognition under Clutter and Occlusion

Article Type: Full length article

Keywords: Keywords: Local surface feature; 3D object recognition; Point-cloud; Feature description; Clutter; Occlusion

Corresponding Author: Mr. Yulan Guo,

Corresponding Author's Institution: National University of Defense Technology

First Author: Yulan Guo

Order of Authors: Yulan Guo; Ferdous Sohel, Ph.D.; Mohammed Bennamoun, Ph.D.; Jianwei Wan, Ph.D.; Min Lu, Ph.D.

A Novel Local Surface Feature for 3D Object Recognition under Clutter and Occlusion

Yulan Guo[a,b,*], Ferdous Sohel[b], Mohammed Bennamoun[b], Jianwei Wan[a], Min Lu[a]

[a] *College of Electronic Science and Engineering, National University of Defense Technology, Changsha, 410073, P.R.China*
[b] *School of Computer Science and Software Engineering, The University of Western Australia, 35 Stirling Highway, 6009, Australia.*

Abstract

This paper presents a highly distinctive local surface feature called the TriSI feature for recognizing 3D objects in the presence of clutter and occlusion. For a feature point, we first construct a unique and repeatable Local Reference Frame (LRF) using the implicit geometrical information of neighboring triangular faces. We then generate three signatures from the three orthogonal coordinate axes of the LRF. These signatures are concatenated and then compressed into a TriSI feature. Finally, we propose an effective 3D object recognition algorithm based on hierarchical feature matching. We tested our TriSI feature on two popular datasets. Rigorous experimental results show that the TriSI feature was highly descriptive and outperformed existing algorithms under all levels of Gaussian noise, Laplacian noise, shot noise, varying mesh resolutions, occlusion, and clutter. Moreover, we tested our TriSI-based 3D object recognition algorithm on four standard datasets. The experimental results show that our algorithm achieved the best overall recognition results on these datasets.

Keywords: Local surface feature; 3D object recognition; Point-cloud; Feature description; Clutter; Occlusion.

1. Introduction

Determining the identities and poses (i.e., positions and orientations) of objects present in a point cloud is the main task of any 3D object recognition system. Three-dimensional object recognition has numerous applications including robotics, biometrics, navigation, remote sensing, medical diagnosis, entertainment, and CAD/CAM engineering [45, 49, 20]. Because of the rapid technological development of 3D imaging sensors, point clouds are becoming more available and accessible. The data availability together with the progress

*Corresponding author
Email address: yulan.guo@nudt.edu.cn (Yulan Guo)

in high-speed computing devices have contributed to the flourishing of research on 3D object recognition algorithms in recent years [19, 16].

Existing 3D object recognition algorithms can broadly be divided into two categories: global feature-based and local feature-based. Global feature-based algorithms extract features from the whole input object/scene. They have achieved promising results in the areas of 3D model retrieval and shape classification [13, 14]. Examples of this type of algorithm include Shape Distribution [34], Spatial Structure Circular Descriptor [11], and view-based methods [13, 12, 15, 38]. They are, however, sensitive to occlusion and clutter and require the objects of interests to be segmented beforehand from the background. In contrast, local feature-based algorithms are the major focus of research in 3D object recognition because they are more robust to occlusion and clutter compared to their global counterparts. In a local feature-based algorithm, local features are first extracted from each model and stored in the library. The algorithm then extracts a set of local features from an input scene and matches them against the model features to yield feature correspondences [24]. These feature correspondences are then used to generate candidate models and transformation hypotheses, which are finally verified to obtain the identity and pose of the object in the scene. In the process of object recognition, the descriptiveness and robustness of the local surface features play a significant role [40]. A feature should, therefore, be descriptive and robust to a set of nuisances, including noise, varying mesh resolutions, occlusion, and clutter.

A number of local surface features have been proposed in the literature (the reader should refer to [20] for a comprehensive review). Some examples are the Point Signature [4], spin image [23], 3D Shape Context (3DSC) [10], tensor [31], Variable-Dimensional Local Shape Descriptors (VD-LSD) [40], Exponential Map (EM) [1], Signature of Histograms of OrienTations (SHOT) [42], and Rotational Projection Statistics (RoPS) [17]. However, most of the existing local surface features suffer from either non-uniqueness, low descriptiveness, weak robustness to noise, or high sensitivity to varying mesh resolutions [1] (see more details in Section 2).

To address these limitations, we propose a highly discriminative and robust local surface feature named Tri-Spin-Image (TriSI). We first build a unique and repeatable Local Reference Frame (LRF) for each feature point to achieve the invariance with respect to rigid transformations. We then generate three signatures to encode the geometrical information of the local surface around three different reference axes. The signatures are then concatenated to form a raw TriSI feature vector. The feature is further compressed using the Principal Component Analysis (PCA) technique. The performance of the TriSI feature was tested on two popular datasets, namely, the Bologna Dataset 1 (BoD1) [42] and the UWA 3D Object Recognition Dataset (U3OR) [30]. The experimental results show that the TriSI feature is very descriptive in terms of precision and recall. It is also very robust with respect to noise, varying mesh resolutions, occlusion, and clutter. We also develop a hierarchical 3D object recognition algorithm. The performance of the object recognition algorithm was evaluated on four standard datasets (i.e., BoD1 [42], U3OR [30], the Queen's LIDAR Dataset

(QuLD) [40], and the Ca' Fascari Venezia Dataset (CFVD) [35]). These datasets contain several nuisances including various object poses, different imaging techniques, noise, varying mesh resolutions, occlusion, and clutter. The experimental results show that the TriSI-based algorithm outperformed the state-of-the-art algorithms, such as the spin image-, SHOT-, 3DSC-, tensor-, VD-LSD-, EM-, and RoPS-based algorithms.

The rest of this paper is organized as follows: Section 2 introduces related work. Section 3 describes the proposed TriSI feature and 3D object recognition algorithm. Section 4 presents the feature matching results on two popular datasets. Section 5 presents the object recognition results on four standard datasets. Section 6 presents a summary and discussion. Section 7 concludes this paper.

2. Related Work

Descriptiveness and robustness are two important requirements for a qualified local surface feature [17]. Because a point cloud is an explicit representation of the 3D surface of an object/scene, it is rational to describe a local surface by generating histograms according to the spatial distributions of the neighboring points. Several features following this scheme have been proposed, which are described in detail in this section.

One of the most popular local surface features is the spin image [23]. It first uses the normal of a feature point as the reference axis and encodes each neighboring point of the feature point by two parameters. It then accumulates the number of neighboring points into a 2D histogram according to these two parameters. The spin image is robust to occlusion and clutter [23]. However, because the cylindrical angular information of the neighboring points is discarded when projecting these points from a 3D space onto a 2D space, the discriminating power of the spin image is limited [50]. Moreover, the spin image is sensitive to varying mesh resolutions [32]. Several variants of the original spin image have been proposed, including a spin image with spherical parameterization [22], multi-resolution spin image [8], spherical spin image [36], and scale invariant spin image [6]. Because all of these improved spin image features still project the 3D information of the neighboring points onto a single 2D map, their descriptiveness cannot be significantly improved.

To achieve relatively higher descriptiveness, several methods were proposed to encode the local surface information by accumulating geometric or topological measurements (e.g., point numbers, mesh areas) into a 3D histogram rather than a 2D histogram. Frome et al. proposed a 3DSC feature [10]. It has proved to perform better than spin images for 3D object recognition [10, 40]. However, its application is significantly limited because of its uncertainty in the rotation around the surface normal. To cope with this limitation, Tombari et al. proposed a Unique Shape Context (USC) feature by assigning a unique LRF to each 3DSC feature [43]. The experimental results show that the USC reduces the memory requirement of 3DSC with an improvement in its feature matching performance. Sukno et al. [39] also proposed an Asymmetry Patterns Shape

Context (APSC) to make the descriptor invariant to rotations. Another extension of the shape context feature is the Intrinsic Shape Context (ISC) [25], which is invariant to isometric deformations. Beyond these shape context features, Mian et al. proposed a tensor representation by aggregating the surface areas into a set of 3D grids. The tensor feature attained a better performance for 3D object recognition compared to the spin image [30]. However, the dimensionality of a 3D tensor is too high to achieve efficient feature matching. Further, Zhong developed an Intrinsic Shape Signature (ISS) by counting the number of points falling into each grid of a uniformly and homogeneously divided spherical angular space [50]. ISS outperformed the spin image and 3DSC in the presence of noise, occlusion, and clutter.

Taati and Greenspan [40] first extracted a set of invariant properties (including position, direction and dispersion properties) for each point of a point cloud. They then performed histogramming on the invariant properties of neighboring points of a feature point to generate a VD-LSD feature. Two histogramming schemes (i.e., scalar quantization and vector quantization) are used to obtain the VD-LSD(SQ) and VD-LSD(VQ) features, respectively. VD-LSD is a generalization for a large class of features including the spin image, point signature, and tensor. VD-LSD requires a training process to learn the optimal subset of properties for the feature generation of a particular object. Recently, Guo et al. [17] proposed a RoPS feature by rotationally projecting the neighboring points of a feature point onto three 2D planes and by calculating a set of statistics for the distributions of these projected points. The experimental results showed that RoPS outperformed the state-of-the-art algorithms, including SHOT, spin image, and VD-LSD.

Achieving both a high power of descriptiveness and strong robustness to various nuisances is still a challenging task faced by existing methods. We, therefore, propose a novel TriSI feature that simultaneously satisfies all of these requirements (as demonstrated in Section 4). Note that while our proposed TriSI feature is somewhat influenced by the spin image concept, the major difference between the TriSI feature and spin image is twofold. First, we use a robust 3D LRF rather than a normal vector to generate spin images. Because the calculation of a normal vector requires surface differentials, it is relatively sensitive to noise. In contrast, our LRF does not rely on any surface differential. It is very robust to both noise and varying mesh resolutions. Second, we generate three signatures around the three axes of the LRF rather than just a single spin image around the normal. Consequently, the proposed TriSI feature is more discriminative compared to the spin image because the former encodes more information of the local surface.

Similar to [17, 10, 23, 30], the proposed algorithm follows a scheme, which consists of three key steps (i.e., LRF construction, feature generation, and object recognition). The LRF proposed in this paper is based on the one presented in [17]. However, the underlying techniques used in this paper are different and more advanced compared with all of these other techniques (including our previous work in [17]). First, the LRF is improved, and different weight combinations are investigated to select the optimal parameters (Sections 3.1.1 and 4.1.1). The

experimental results show that the improved LRF is more robust to shot noise compared to the LRF in [17]. Second, the TriSI feature is generated in a completely different way compared to [17] (Section 3.1.2). Rather than rotating the local surface and encoding the distributions of a set of projected points on the coordinate planes (as in [17]), the TriSI feature succinctly concatenates the signatures that represent the point distribution in three cylindrical coordinate systems. Third, the proposed 3D object recognition algorithm is more adaptive to parameter settings by using a hierarchical feature matching strategy (Section 3.2). Comparative results clearly demonstrate that the proposed algorithm outperforms our previous work [17] on all datasets in terms of both feature matching (Section 4) and object recognition (Section 5).

The main contributions of this paper are as follows:

(i) We propose a discriminative and robust TriSI feature as an improvement of the method previously presented in [17]. More specifically, the improvement consists of an improved LRF and a new 3D local surface feature. The experimental results show that our TriSI feature outperformed existing features including spin image, SHOT, and RoPS by a large margin in terms of recall and precision. The TriSI feature is also very robust to noise, varying mesh resolutions, occlusion, and clutter.

(ii) We propose a hierarchical 3D object recognition algorithm. The experimental results show that the TriSI-based 3D object recognition algorithm achieved the best recognition performance on all of these datasets when compared with a number of existing techniques.

3. TriSI-based 3D Object Recognition

3.1. TriSI Feature

A 3D local feature should be invariant to rigid transformations including rotations and translations. A unique and repeatable LRF is, therefore, adopted to represent the local surface in a pose-invariant local coordinate system rather than a sensor-centered coordinate system (Section 3.1.1). A local feature should also be highly descriptive, which is achieved by encoding the information of a local surface around three orthogonal axes (Section 3.1.2). For the sake of efficient feature matching, a feature should be compact. This is achieved by performing the PCA transform on the feature vector and by extracting the most significant components of the feature (Section 3.1.3). Consequently, the process of generating a compact TriSI feature includes LRF construction, TriSI generation, and TriSI compression.

3.1.1. LRF Construction

Our LRF is based on our previously presented LRF in [17]. However, [17] did not fully consider the cases of outliers and shot noise, which led to unstable LRFs in these circumstances. In this paper, we enhance the LRF by considering a different weighting strategy, which is more robust than the one described

in [17]. For the sake of completeness, we describe below the LRF, which was originally introduced in [17], and the improvements made in this paper.

Given a point cloud $\mathcal{P} \in \mathbb{R}^3$, which represents the local surface, it is first converted into a triangular mesh $\mathcal{S}$ [30]. The mesh $\mathcal{S}$ consists of N_p vertices and N_f triangular faces. For a given feature point $\boldsymbol{p}$ and its support radius r, a local triangular mesh $\mathcal{L}$ is cropped from $\mathcal{S}$ such that all vertices in $\mathcal{L}$ are within the distance of r from point $\boldsymbol{p}$. Assume that $\mathcal{L}$ contains n_p vertices and n_f triangles with the ith triangle consisting of three vertices $\boldsymbol{q}_{i1}$, $\boldsymbol{q}_{i2}$ and $\boldsymbol{q}_{i3}$. It is possible to derive a scatter matrix of all points lying on the local surface $\mathcal{L}$, including "invisible" points within a triangle (interpolated points on the surface of the triangle). For any invisible point $\boldsymbol{q}_i(\gamma_1, \gamma_2)$ in the ith triangle, it can be expressed with the three vertices $\boldsymbol{q}_{i1}$, $\boldsymbol{q}_{i2}$ and $\boldsymbol{q}_{i3}$ as:

$$\boldsymbol{q}_i(\gamma_1, \gamma_2) = \boldsymbol{q}_{i1} + \gamma_1(\boldsymbol{q}_{i2} - \boldsymbol{q}_{i1}) + \gamma_2(\boldsymbol{q}_{i3} - \boldsymbol{q}_{i1}). \tag{1}$$

The scatter matrix $\mathbf{C}_i$ of all points in the ith triangle can therefore be calculated through an integral over this triangular face. That is,

$$\begin{aligned}
\mathbf{C}_i &= \int_0^1 \int_0^{1-\gamma_2} (\boldsymbol{q}_i(\gamma_1, \gamma_2) - \boldsymbol{p})(\boldsymbol{q}_i(\gamma_1, \gamma_2) - \boldsymbol{p})^{\mathrm{T}} d\gamma_1 d\gamma_2 \\
&= \frac{1}{12} \sum_{m=1}^{3} \sum_{n=1}^{3} (\boldsymbol{q}_{im} - \boldsymbol{p})(\boldsymbol{q}_{in} - \boldsymbol{p})^{\mathrm{T}} \qquad (2) \\
&\quad + \frac{1}{12} \sum_{m=1}^{3} (\boldsymbol{q}_{im} - \boldsymbol{p})(\boldsymbol{q}_{im} - \boldsymbol{p})^{\mathrm{T}}. \qquad (3)
\end{aligned}$$

The overall scatter matrix $\mathbf{C}$ of the local surface $\mathcal{L}$ is then calculated as a weighted sum of the scatter matrices of all individual triangles on $\mathcal{L}$. That is,

$$\mathbf{C} = \sum_{i=1}^{n_f} \omega_{i1} \omega_{i2} \mathbf{C}_i, \tag{4}$$

where the weight ω_{i1} is proportional to the area a_i of the ith triangle related to the whole area of the local surface $\mathcal{L}$. That is,

$$w_{i1} = \frac{a_i^{\kappa_1}}{\sum_{i=1}^{n_f} a_i^{\kappa_1}}. \tag{5}$$

The weight ω_{i2} is proportional to the distance from the centroid of the ith triangle to the feature point $\boldsymbol{p}$. That is,

$$w_{i2} = \left(r - \left\| \boldsymbol{p} - \frac{\boldsymbol{q}_{i1} + \boldsymbol{q}_{i2} + \boldsymbol{q}_{i3}}{3} \right\| \right)^{\kappa_2}. \tag{6}$$

where $k_1 \in \mathbb{Z}^+$, $k_2 \in \mathbb{Z}^+$. k_1 and k_2 are used to control the relative weight of each individual triangle according to its area and the distance to the feature point, respectively. Consequently, different parameters κ_1 and κ_2 result in different

weighting strategies. The selection of the two parameters is further investigated in Section 4.1.1. To address irregular triangles (e.g., caused by outliers and shot noise), an outlier-rejection technique is proposed. That is, the weight ω_{i1} is set to 0 if one or more edges in the ith triangle are longer than τ_e times the mesh resolution. Based on an experiment and observations of several real-life datasets, τ_e is chosen as five in this paper. Note that irregular triangles cannot be handled well using the LRF in [17] because no such technique is adopted.

We then perform an eigenvalue decomposition on $\mathbf{C}$:

$$\mathbf{CV} = \mathbf{VD}, \tag{7}$$

where $\mathbf{D}$ is a diagonal matrix with diagonal entries equal to the eigenvalues of $\mathbf{C}$ and $\mathbf{V}$ is a matrix with columns equal to the eigenvectors of $\mathbf{C}$. These eigenvectors $\boldsymbol{v}_1$, $\boldsymbol{v}_2$, and $\boldsymbol{v}_3$ are ordered according to their associated eigenvalues such that $\boldsymbol{v}_1$ corresponds to the largest eigenvalue. Because $\boldsymbol{v}_1$, $\boldsymbol{v}_2$, and $\boldsymbol{v}_3$ are orthogonal, they can be used as the basis of an LRF. However, their directions are ambiguous. That is, $-\boldsymbol{v}_1$, $-\boldsymbol{v}_2$, and $-\boldsymbol{v}_3$ are also eigenvectors of the scatter matrix $\mathbf{C}$. We therefore, adopt a sign disambiguation technique.

For any asymmetric surface, the sign of an eigenvector $\boldsymbol{v}_k\,(k=1,3)$ can be determined by calculating the scalar products between the eigenvector $\boldsymbol{v}_k$ and the vectors from the feature point $\boldsymbol{p}$ to all points $\boldsymbol{q}_i\,(\gamma_1,\gamma_2)\,(i=1,2,\ldots,n_f)$ on the local surface $\mathcal{L}$. If the majority of the scalar products are positive, the sign of $\boldsymbol{v}_k$ remains unchanged. Otherwise, the sign of $\boldsymbol{v}_k$ is inverted. Specifically, the sign of the eigenvector $\boldsymbol{v}_k\,(k=1,3)$ is defined as:

$$\begin{aligned} \Lambda_k &= \operatorname{sgn}\left(\sum_{i=1}^{n_f}\omega_{i1}\omega_{i2}\int_0^1\int_0^{1-\gamma_2}\left(\boldsymbol{q}_i\left(\gamma_1,\gamma_2\right)-\boldsymbol{p}\right)\boldsymbol{v}_k d\gamma_1 d\gamma_2\right) \\ &= \operatorname{sign}\left(\sum_{i=1}^{n_f}\omega_{i1}\omega_{i2}\left(\sum_{m=1}^{3}\left(\boldsymbol{q}_{im}-\boldsymbol{p}\right)\boldsymbol{v}_k\right)\right). \end{aligned} \tag{8}$$

where, the function $\operatorname{sgn}(\cdot)$ returns a value of +1 for a positive number and -1 for a negative number. Therefore, two unambiguous vectors $\widetilde{\boldsymbol{v}_1}$ and $\widetilde{\boldsymbol{v}_3}$ are obtained by:

$$\widetilde{\boldsymbol{v}_k} = \Lambda_k \boldsymbol{v}_k. \tag{9}$$

Given $\widetilde{\boldsymbol{v}_1}$ and $\widetilde{\boldsymbol{v}_3}$, the vector $\widetilde{\boldsymbol{v}_2}$ is then defined as $\widetilde{\boldsymbol{v}_3}\times\widetilde{\boldsymbol{v}_1}$. Consequently, the position of feature point $\boldsymbol{p}$ and the three unambiguous vectors $\{\widetilde{\boldsymbol{v}_1},\widetilde{\boldsymbol{v}_2},\widetilde{\boldsymbol{v}_3}\}$ constitute an LRF for point $\boldsymbol{p}$.

It is commonly known that the traditional PCA is highly sensitive to noise and outliers [7, 46]; even a single corrupted point can significantly alter the results. A number of algorithms have been proposed in the literature to improve the robustness of the traditional PCA using the techniques of outlier pursuit [46], M-estimation [7], and convex optimization [3]. In contrast, our proposed LRF is generated by a weighted continuous PCA technique with outlier rejection. The experimental results show that it is sufficiently robust to both noise and outliers (Section 4.2).

3.1.2. TriSI Generation

Given a feature point $\boldsymbol{p}$, the local surface $\mathcal{L}$ and its LRF vectors $\{\widetilde{\boldsymbol{v}_1}, \widetilde{\boldsymbol{v}_2}, \widetilde{\boldsymbol{v}_3}\}$, the information from this local surface is encoded by three signatures $\{\mathbf{SI}_1, \mathbf{SI}_2, \mathbf{SI}_3\}$ which are generated around three axes. Because the three axes are orthogonal to each other, the information from the three signatures is complementary and relatively irredundant. Therefore, the resulting TriSI feature is highly descriptive. An illustration of the generation of a TriSI feature is shown in Fig. 1.

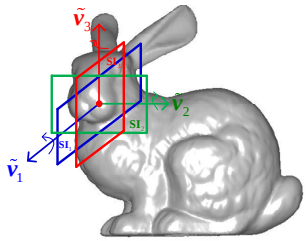


Figure 1: An illustration of the generation of a TriSI feature (Figure best seen in color).

We generate a signature $\mathbf{SI}_1$ by encoding the point distribution around the $\widetilde{\boldsymbol{v}_1}$ axis. Given a point $\boldsymbol{q} \in \mathbb{R}^3$ on the local surface $\mathcal{L}$, we use a function $f : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ to map its 3D coordinate to a 2D space. The 2D space is represented by two parameters α and β. Here, α is the perpendicular distance of $\boldsymbol{q}$ from the line that passes through $\boldsymbol{p}$ and is parallel to $\widetilde{\boldsymbol{v}_1}$, and β is the signed perpendicular distance to the plane that goes through $\boldsymbol{p}$ and is perpendicular to $\widetilde{\boldsymbol{v}_1}$. That is,

$$\alpha = \sqrt{\|\boldsymbol{q} - \boldsymbol{p}\|^2 - (\widetilde{\boldsymbol{v}_1} \cdot (\boldsymbol{q} - \boldsymbol{p}))^2}, \tag{10}$$

$$\beta = \widetilde{\boldsymbol{v}_1} \cdot (\boldsymbol{q} - \boldsymbol{p}) \,. \tag{11}$$

Once all points on $\mathcal{L}$ are represented by the two parameters (α, β), we discretize the 2D space (α, β) by $b \times b$ bins. We count the number of surface points falling into each bin, which results in a 2D histogram. In fact, this 2D histogram records the distribution of points in a cylindrical coordinate system with $\widetilde{\boldsymbol{v}_1}$ as its reference axis. Because the 2D histogram encodes the information for the relative position of 3D points lying on the local surface $\mathcal{L}$, part of the 3D metric information is preserved, and the shape of the local surface is presented. To make the resulting feature less sensitive to the position variations of the points (e.g., because of noise and different viewpoints), the 2D histogram is further bi-linearly interpolated, resulting in a final signature $\mathbf{SI}_1$. Because $\mathbf{SI}_1$ is generated with respect to the feature point $\boldsymbol{p}$ and its intrinsic direction $\widetilde{\boldsymbol{v}_1}$, the signature is invariant to rigid transformations. Note that a spin image feature [23] is generated by encoding the point distribution around the surface normal $\boldsymbol{n}$ at point $\boldsymbol{p}$ rather than the axis $\widetilde{\boldsymbol{v}_1}$, while the remaining process is the same as $\mathbf{SI}_1$. Therefore, most variants [8, 36] of the original spin image can be seamlessly integrated with our method to obtain various TriSI features.

The signature $\mathbf{SI}_1$ generated from a single reference axis is insufficient to represent the rich information of a local surface. We therefore create two other signatures $\mathbf{SI}_2$ and $\mathbf{SI}_3$ by following a similar method we adopted to produce $\mathbf{SI}_1$. That is, $\mathbf{SI}_2$ and $\mathbf{SI}_3$ are generated by substituting the $\widetilde{\boldsymbol{v}_1}$ in equations (10-11) with $\widetilde{\boldsymbol{v}_2}$ and $\widetilde{\boldsymbol{v}_3}$, respectively. These three signatures encode the information of the local surface $\mathcal{L}$ from three orthogonal axes. To produce a relatively high discriminative feature, the three signatures are concatenated to obtain a raw TriSI feature, that is,

$$\boldsymbol{f} = \{\mathbf{SI}_1, \mathbf{SI}_2, \mathbf{SI}_3\}. \tag{12}$$

3.1.3. TriSI Compression

To make the feature vector more compact, the raw TriSI feature is further compressed by projecting it onto a PCA subspace. The PCA subspace is learned from a set of training feature vectors $\{\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_{n_t}\}$, where n_t is the total number of training features. The PCA algorithm converts a set of possibly correlated features into a set of values for linearly uncorrelated variables (i.e., principal components).

Given the training features $\{\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_{n_t}\}$, the covariance matrix $\mathbf{M}$ is calculated as:

$$\mathbf{M} = \sum_{i=1}^{n_t} \left(\boldsymbol{f}_i - \overline{\boldsymbol{f}}\right) \left(\boldsymbol{f}_i - \overline{\boldsymbol{f}}\right)^{\mathrm{T}}, \tag{13}$$

where,

$$\overline{\boldsymbol{f}} = \frac{1}{n_t} \sum_{i=1}^{n_t} \boldsymbol{f}_i. \tag{14}$$

The eigenvalue decomposition is then applied to $\mathbf{M}$:

$$\mathbf{MV} = \mathbf{VD}, \tag{15}$$

where $\mathbf{D}$ is a diagonal matrix with diagonal entries equal to the eigenvalues of $\mathbf{M}$ and $\mathbf{V}$ is a matrix with columns equal to the eigenvectors of $\mathbf{M}$. The PCA subspace is constructed by using the eigenvectors, which correspond to the n_{sf} largest eigenvalues. The value of n_{sf} is chosen such that ϑ of the fidelity of the training features is preserved in the compressed features (as further discussed in Section 4.1.3).

Therefore, the compressed vector $\widehat{\boldsymbol{f}_i}$ of a feature $\boldsymbol{f}_i$ is:

$$\widehat{\boldsymbol{f}_i} = \mathbf{V}_{n_{sf}}^{\mathrm{T}} \boldsymbol{f}_i, \tag{16}$$

where $\mathbf{V}_{n_{sf}}^{\mathrm{T}}$ is the transpose of the first n_{sf} columns of $\mathbf{V}$.

3.2. 3D Object Recognition

Our 3D object recognition algorithm follows the most common recognition scheme [40, 41, 17, 20], and it consists of four modules: offline preprocessing, feature generation, feature matching, and hypothesis verification. The block diagram of the 3D Object recognition algorithm is shown in Fig. 2.

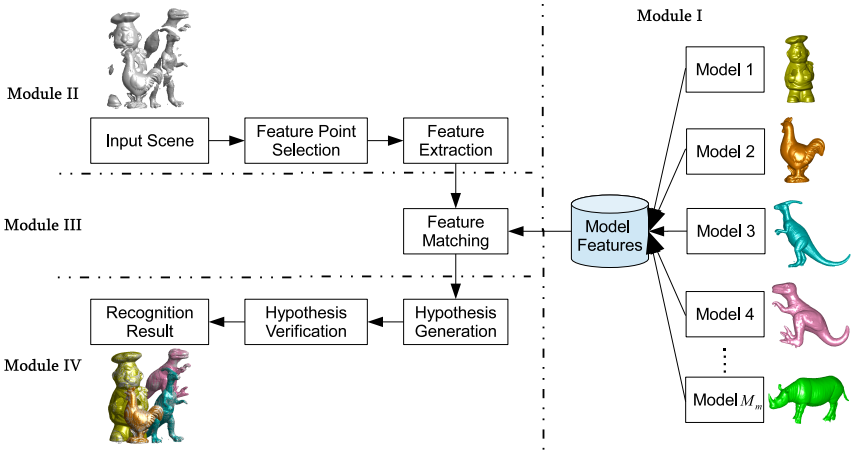


Figure 2: Block diagram of TriSI based 3D Object recognition algorithm

Module I-Offline Preprocessing: N_m feature points are first uniformly selected from each model $\mathcal{M}_m$. Next, N_m model features $\{\boldsymbol{f}_1^m, \boldsymbol{f}_2^m, \cdots, \boldsymbol{f}_{N_m}^m\}$ are obtained from the model $\mathcal{M}_m$ by extracting a TriSI feature $\boldsymbol{f}_i^m$ at each feature point. These features from all models are used to construct a PCA subspace $\mathbf{V}_n^{\mathrm{T}}$. Then, each feature $\boldsymbol{f}_i^m$ is projected onto the PCA subspace to obtain a compressed feature $\widehat{\boldsymbol{f}_i^m}$ using equation 16. These compressed features are indexed and stored in the library.

Module II-Feature Generation: Given an input scene $\mathcal{S}$, N_s feature points are uniformly selected. For each feature point, an LRF is defined, and a TriSI feature $\boldsymbol{f}_i^s$ is then generated. The feature is then projected onto the PCA subspace to obtain the compressed TriSI feature $\widehat{\boldsymbol{f}_i^s}$ using equation 16.

Module III-Feature Matching: For each scene feature $\widehat{\boldsymbol{f}_i^s}$, the nearest and second nearest distances between $\widehat{\boldsymbol{f}_i^s}$ and the stored model features are calculated. If the model feature $\widehat{\boldsymbol{f}_i^m}$ is the nearest neighbor to $\widehat{\boldsymbol{f}_i^s}$ and the ratio of the nearest distance to the second nearest distance is less than a threshold τ, the scene feature $\widehat{\boldsymbol{f}_i^s}$ and model feature $\widehat{\boldsymbol{f}_i^m}$ are considered a feature correspondence $\left(\widehat{\boldsymbol{f}_i^s}, \widehat{\boldsymbol{f}_i^m}\right)$ [28]. It is demonstrated that the adopted nearest neighbor distance ratio-based feature matching strategy outperforms the nearest neighbor based-strategy because the former additionally penalizes the features, which have several similar matches [28, 33]. Then, each feature correspondence $\left(\widehat{\boldsymbol{f}_i^s}, \widehat{\boldsymbol{f}_i^m}\right)$ gives

one vote to the mth model $\mathcal{M}_m$. It also provides a transformation estimate $\mathbf{T}_i^m$ between the input scene $\mathcal{S}$ and model $\mathcal{M}_m$ by aligning their LRFs.

Module IV-Hypothesis Verification: The models are sorted according to their associated number of feature correspondences (i.e., votes) and are verified by turn. For each transformation estimate $\mathbf{T}_i^m$ between $\mathcal{S}$ and the candidate model $\mathcal{M}_m$, all transformation estimates that are close to $\mathbf{T}_i^m$ are retrieved. These retrieved estimates form a cluster of consistent transformation estimates. The cluster center provides the final transformation $\widehat{\mathbf{T}}_i^m$ for the feature correspondence $\left(\widehat{\boldsymbol{f}_i^s}, \widehat{\boldsymbol{f}_i^m}\right)$. Note that the dimension of each cluster center is 6DOF because the dimensions of the rotation and translation are 3DOF each. A confidence score is then assigned to each cluster based on the number of members in the cluster and their feature distances. The transformations with confidence scores larger than half of the maximum score are elected as transformation candidates. The candidate model $\mathcal{M}_m$ is then aligned with the input scene $\mathcal{S}$ using each transformation candidate. If $\mathcal{M}_m$ is accurately aligned with a portion of $\mathcal{S}$, the candidate model is accepted, and the scene points that are close to the candidate model are segmented from the scene (for more details on alignment and verification, the author should refer to [17]). Otherwise, the transformation candidate is rejected, and the next transformation candidate is verified by turn. If all of the transformation candidates between $\mathcal{S}$ and $\mathcal{M}_m$ have already been verified, the next candidate model is verified using the same approach. This verification procedure continues until all of the candidate models have been verified or too few points are left in the scene $\mathcal{S}$. Beyond pose clustering, other techniques such as the constrained interpretation tree [1], Random Sample Consensus (RANSAC) [9, 40], and geometric consistency [23] can also be used to generate transformation hypotheses for verification (for a comprehensive review, the author should refer to [20]).

Most existing object recognition algorithms [28, 17] use a fixed threshold τ for feature matching (Module III). It is, however, very challenging to select the most appropriate threshold for a specific application. On one hand, although the produced matching results using a low threshold are accurate, the object recognition rate is low because the recall of the feature matching is low. On the other hand, a large threshold increases the recognition accuracy at the expense of a high computational cost. This is because more hypotheses (which are generated from the feature matching phase) need to be verified in the subsequent steps. In this paper, we propose a hierarchical matching strategy for effective and efficient 3D object recognition. Specifically, we first use a small threshold τ to perform feature matching (Module III) and hypothesis verification (Module IV), and the recognized objects are segmented from the scene. Meanwhile, the scene features that are associated with the recognized objects are removed. Next, we increase the threshold τ and perform feature matching (Module III) using the remaining scene features. This is subsequently followed by hypothesis verification (Module IV) which works on the remaining scene points. The aforementioned feature matching and hypothesis verification processes are repeated with a set of thresholds. In this paper, we increase the threshold from 0.7 to

1 with an increment step of 0.1. The minimum threshold is set to 0.7 because too few feature correspondences can be produced with a low threshold of less than 0.7. The maximum threshold is set to one because no ratio of the nearest distance to the second nearest distance can be larger than one.

The advantage of the proposed hierarchical matching strategy is threefold. **First**, the efficiency of the object recognition algorithm is improved by starting with a low threshold. That is, a low threshold (e.g., τ=0.7) results in a relatively small number of feature correspondences. These correspondences are very reliable, and the resulting model hypotheses are likely to be present in the input scene. **Second**, the accuracy of the object recognition algorithm is boosted by subsequently using a set of thresholds. That is, a large threshold (e.g., τ=1) results in a large number of feature correspondences and produces more plausible hypotheses for verification. Because our algorithm starts from the lowest threshold, most of the input scene can be recognized and segmented efficiently by the processes that are based on low thresholds, leaving only a small number of points to the subsequent processes (which are based on large thresholds). **Third**, the proposed algorithm is more general because it adaptively uses a range of thresholds rather than a fixed threshold. However, with the existing algorithms, the fixed threshold used may be dependent on and only appropriate for a specific dataset.

4. Evaluation of the TriSI Feature

The parameters for TriSI feature generation were first trained on an independent tuning dataset (Section 4.1). The TriSI feature was then tested on two popular datasets (Sections 4.2 and 4.3), namely BoD1 [42] and U3OR [30]. The performance of feature matching was evaluated with the frequently used *Recall* versus 1-*Precision* Curve (RPC) [33]. Ideally, an RPC should lie at the top left corner area of a plot with a high recall and high precision. The performance of TriSI was compared with the spin image [23], SHOT [42], and RoPS [17] with respect to a set of nuisances including Gaussian noise, Laplacian noise, shot noise, varying mesh resolutions, occlusion, and clutter.

4.1. Selection of the Parameters

The tuning dataset contains six models and six scenes. The models were obtained from the Stanford 3D Scanning Repository [5], and the scenes were created by downsampling each model to half of its original mesh resolution (mr) and adding both Gaussian noise (with a standard deviation of 0.1 mr) and shot noise (with a outlier ratio of 0.2%). For each scene and its corresponding model, we first randomly selected 1000 feature points from the model. The scene and model were then automatically aligned using the ground-truth transformation, and the closest points in the aligned scene were selected. Each feature point in the model and its closest feature point in the scene constitute a ground-truth point correspondence. We tested the LRF with different weight combinations (i.e., κ_1 and κ_2). We also tested the TriSI feature with different bin numbers b

and fidelity percentages ϑ. The optimal parameters were selected by the tuning experiments.

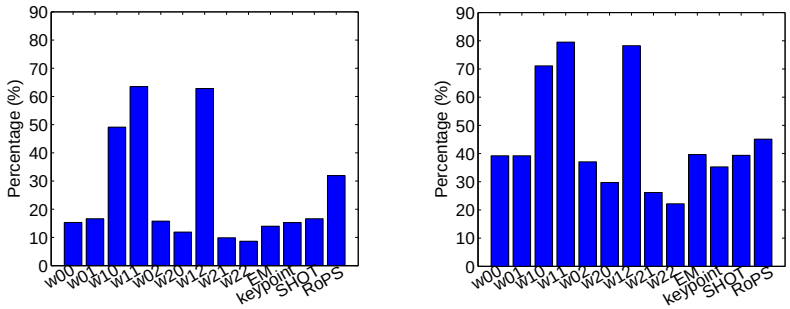
For any local surface feature, the support radius r is an important parameter, which determines the amount of local surface that is described by the feature descriptor. A large support radius provides more descriptiveness at the cost of a higher sensitivity to occlusion and clutter. Based on our previous work [17], in this paper, we set the support radius r to 15 mr for the generation of the local feature as a compromise between the matching accuracy and robustness. Note that although a fixed support radius was used in this paper, any adaptive scale detection algorithm can alternatively be integrated with our TriSI feature. However, the work of feature scale detection is out of the scope of this paper and our focus is on the quality of the feature description rather than feature detection. For more details on feature position and scale detection, the reader should refer to the review and evaluation papers in [44, 20].

4.1.1. LRF

We used nine different weight combinations w_{mn} $(m, n = 0, 1, 2)$ to generate LRFs, where w_{mn} denotes that κ_1=m and κ_2=n. Given a weight combination, we generated a LRF for each of the scene and model points used for a point correspondence. We then calculated the error of this LRF estimation as the rotation angle between the two LRFs [31]. If the error was less than an error threshold, the estimation was considered correct. Finally, we calculated the percentage of correct estimations with respect to the total estimations as our performance measure. The results achieved with the two different error thresholds (i.e., 5°and 10°) are shown in Fig. 3. It can be seen that the ranking of these LRFs is the same under the two different error thresholds. It is also clear that the LRF with κ_1=1 and κ_2=2 achieved the best performance (which was then selected as the optimal combination in the rest of the paper). Moreover, all combinations with κ_1=1 achieved a better performance compared to those with κ_1=0 or κ_1=2. Because the overall scatter matrix is an integral over the point scatter matrices of all individual triangles, it is reasonable that the contribution from each triangle is linearly related to its surface area (i.e., κ_1=1). We also tested the performance of the LRFs used in EM [1], keypoint [32], SHOT [42], and RoPS [17] features, as shown in Fig. 3. It is clear that our LRF with κ_1=1 and κ_2=2 outperformed the state-of-the-art methods by a large margin. With an error threshold of 10°, the percentages of correct estimations for the LRFs in EM, keypoint, SHOT, RoPS, and our LRF are 39.6%, 35.3%, 39.4, 45.1%, and 78.3%, respectively. Note that the proposed LRF significantly improved the LRF used in RoPS [17] in the case of combined Gaussian noise, shot noise, and decimation.

4.1.2. Bin Number Analysis

We further tested the performance of the TriSI feature with respect to different bin numbers b. We set the bin number to 5, 15, 25, 35, 45, and 55. We used the raw TriSI features without PCA compression to perform feature matching.



(a) error threshold = 5° (b) error threshold = 10°

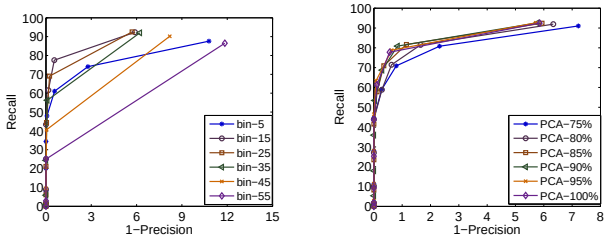
Figure 3: Comparison between different LRFs.

The results are shown in Fig. 4(a). The performance of the TriSI feature improved as the bin number increased from 5 to 15. With a bin number of 15, the TriSI feature achieved the best performance. The performance decreased slightly as the bin number increased further from 15 to 25. For bin numbers larger than 25, the performance of the TriSI feature dropped significantly compared to the results with a bin number of 15. These observations can be explained based on the following two reasons. First, a small bin number (e.g., 5) results in a coarse signature. That is, each bin of the signature encodes information from a large patch of the surface. Consequently, most of the geometrical details are lost, and the resulting TriSI feature lacks descriptiveness. Second, a large bin number (e.g., 55) makes the signature very sensitive to noise and to varying mesh resolutions. Moreover, the signature with a large bin number is also very sparse, and many bins have a value of zero. Consequently, the descriptiveness and robustness of the TriSI feature deteriorates. Moreover, a large bin number will result in an increase in both memory usage (during feature generation) and computational time (during feature matching). We therefore set the bin number to 15 throughout this paper.

4.1.3. PCA Compression

The PCA technique not only transforms the possibly correlated features into uncorrelated variables but also reduces the dimensionality of a TriSI feature [29]. To test the TriSI feature with respect to different levels of compression, we set the fidelity percentage ϑ to 75%, 80%, 85%, 90%, 95%, and 100%. The TriSI features of the six models were used to train the PCA subspace. The feature matching results are shown in Fig. 4(b). It can be seen that the performance increased with the value of ϑ. Specifically, an obvious improvement was achieved as the fidelity percentage ϑ increased from 75% to 95%. Because a small ϑ causes a significant loss of information for the original features, which subsequently deteriorates the performance of the compressed features. The performance for the compressed features with ϑ equal 95% and 100% was nearly the same. This means that the compressed features with a fidelity percentage of 95%

are sufficient and can be considered as a useful representation of their original features. We therefore set the fidelity percentage ϑ to 95% throughout the rest of this paper.



(a) Bin number (b) PCA compression

Figure 4: The performance of TriSI features with different parameters.

4.2. Performance on the BoD1 Dataset

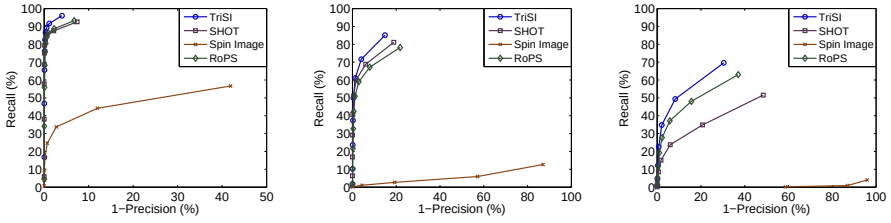
We first tested the performance of our TriSI feature on the BoD1 dataset [42]. The BoD1 dataset contains six models and 45 scenes. The models were taken from the Stanford 3D Scanning Repository [5]. The scenes were generated by randomly placing three to five models to create clutter and pose variances. The ground-truth transformations between each model and its instances in the scenes were provided as a prior. To test the performance of feature matching, 1000 feature points were randomly selected from each model [42]. Their corresponding points in each scene were also obtained to produce ground-truth point correspondences. Note that each pair of corresponding feature points in the scene and models are extracted at approximately the same physical position (i.e., no keypoint localization error is considered). This is the ideal case for feature matching where the influences of keypoint detection and feature description are separated. Then, TriSI, spin image, SHOT, and RoPS features were extracted at each feature point for all scenes and models. The TriSI features were further compressed using the PCA subspace trained on the model features. The lengths for the compressed TriSI, spin image, SHOT, and RoPS features were 29, 225, 320, 135, respectively. Finally, the RPC results for each feature were generated by matching the scene features against the model features [33]. These features were tested with respect to Gaussian noise [42, 48], Laplacian noise, shot noise [48], and varying mesh resolutions [42].

4.2.1. Robustness to Gaussian/Laplacian Noise

To test the performance of these features with respect to Gaussian noise, we added three levels of Gaussian noise with standard deviations of 0.1 mr, 0.3 mr, and 0.5 mr to each scene. For a given standard deviation, Gaussian noise was independently added to the $x-$, $y-$, and $z-$ axes of each scene point (see Fig. 5(a)). The RPC results at different levels of Gaussian noise are shown in Fig. 6.

(a) Gaussian noise (b) Laplacian noise (c) Shot noise (d) Decimation

Figure 5: Scenes with different deformations (Figure best seen in color).



(a) (b) (c)

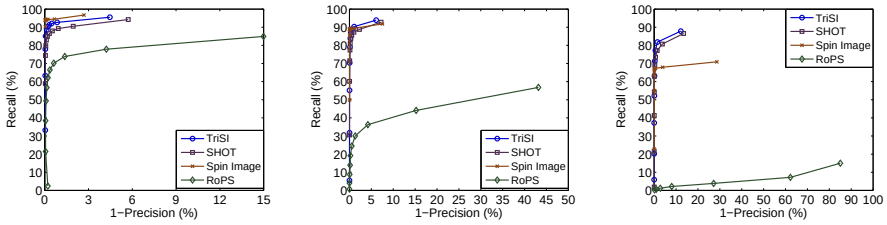
Figure 6: Recall vs 1-precision curves in the presence of Gaussian noise. (a) Standard deviation of 0.1mr. (b) Standard deviation of 0.3mr. (c) Standard deviation of 0.5mr.

The proposed TriSI feature was highly descriptive and robust to Gaussian noise. The TriSI feature achieved the best performance at all levels of Gaussian noise, followed by the RoPS, SHOT and spin image. The TriSI feature achieved the highest recalls of 96%, 85%, and 70% with noise deviations of 0.1mr, 0.3mr, and 0.5mr, respectively. As the deviation of Gaussian noise increased from 0.1mr to 0.5mr, the performance of all four features decreased. However, our TriSI feature was more stable and robust to Gaussian noise compared with the other three features. We also tested the performance of these features with respect to three levels of Laplacian noise (with standard deviations of 0.1mr, 0.3mr, and 0.5mr). The results under Laplacian noise were consistent with those achieved under Gaussian Noise.

4.2.2. Robustness to Shot Noise

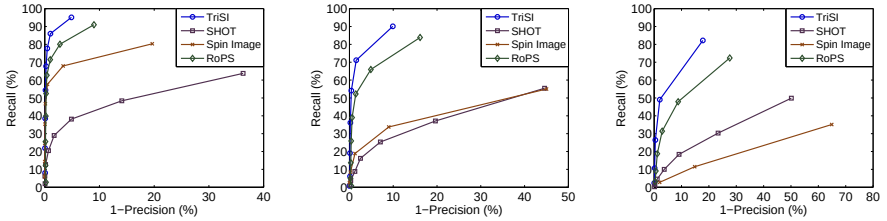
To test the performance of these features with respect to shot noise, we added three levels of shot noise with outlier ratios of 0.2%, 1.0%, and 5.0% to each scene. Given an outlier ratio γ, a ratio γ of the total points in each scene was first selected, and a displacement with an amplitude of 20 mr was then added to each selected point along its normal direction (see Fig. 5(c)), the same as in [48]. The RPC results at different levels of shot noise are shown in Fig. 7.

The TriSI feature was very robust to shot noise. The performance of the TriSI, spin image, and SHOT features was comparable with an outlier ratio of shot noise less than 1.0%. The recall results achieved by all of these features except RoPS were larger than 90%. RoPS is very sensitive to shot noise, and



(a) (b) (c)

Figure 7: Recall vs 1-precision curves in the presence of shot noise. (a) Outlier ratio of 0.2%. (b) Outlier ratio of 1.0%. (c) Outlier ratio of 5.0%.



(a) (b) (c)

Figure 8: Recall vs 1-precision curves with respect to varying mesh resolutions. (a) 1/2 mesh decimation. (b) 1/4 mesh decimation. (c) 1/8 mesh decimation.

its performance dropped sharply as the outlier ratio of shot noise increased. In contrast, the TriSI feature obtained the best results with high recalls of more than 85% when tested on scenes with an outlier ratio of 5.0%. The TriSI feature performed slightly better than the SHOT feature. Note that the scenes with an outlier ratio of 5.0% are very spiky (as illustrated in Fig. 5(c)). Most of the details of these scenes are lost, and their shapes are deformed dramatically from their original shapes. However, our proposed TriSI feature still achieved acceptable results. This clearly indicates that the TriSI feature is very robust to shot noise.

4.2.3. Robustness to Varying Mesh Resolutions

To test the performance of these features with respect to varying mesh resolutions, we resampled each scene to 1/2, 1/4, and 1/8 of its original mesh resolution (see Fig. 5(d)). The RPC results of these features are shown in Fig. 8.

It is clear that the proposed TriSI feature was very robust to varying mesh resolutions. It outperformed the other features by a large margin under all levels of mesh decimation with RoPS in the second position. The TriSI feature achieved a high recall of approximately 95% under 1/2 mesh decimation, as shown in Fig. 8(a). It also achieved a recall of approximately 90% under 1/4 mesh decimation, as shown in Fig. 8(b). The TriSI feature consistently achieved good performance even under 1/8 mesh decimation with a recall of more than 80%, as shown in Fig. 8(c). It is worth noting that the performance of the TriSI

feature under 1/8 mesh decimation was even better than the performance of the spin image under 1/2 mesh decimation, as shown in Figs. 8(a) and (c).

4.2.4. Overall Performance

To test the overall performance of these features, we first resampled each scene to 1/2 of its original mesh resolution. We then added Gaussian noise (with a standard deviation of 0.1 mr) and shot noise (with an outlier ratio of 0.2%) to each scene. The RPC results of the TriSI, spin image, SHOT, and RoPS features are shown in Fig 9. It is clear that the TriSI feature outperformed spin image, SHOT, and RoPS in the presence of Gaussian noise, shot noise, and mesh decimation.

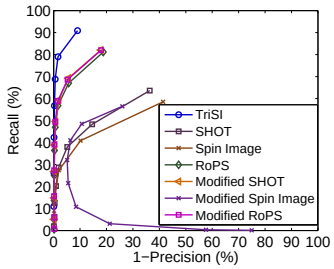


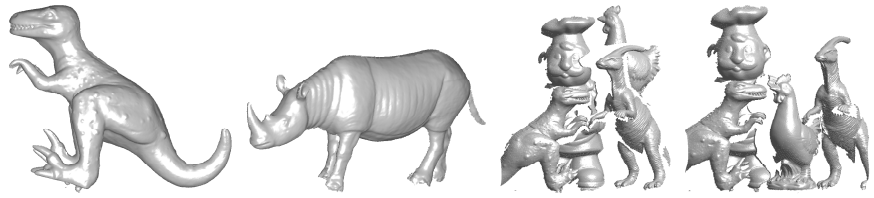
Figure 9: Recall vs 1-precision curve with respect to combined noise and mesh decimation.

To further investigate the individual effectiveness of the proposed LRF and the TriSI feature, we introduced modified spin image, modified SHOT, and modified RoPS features by substituting their original LRFs with our LRF (the same as TriSI). For the generation of the modified spin image, the axis $\widetilde{v_3}$ of our LRF is used as its reference axis. The RPC results of these modified features are shown in Fig 9. The performance of all of these modified features was improved compared with their original counterparts. However, their performance was still inferior to the performance of the TriSI feature. The TriSI feature achieved the best performance, followed by the modified RoPS and modified SHOT features.

It can be concluded that the superior performance of the TriSI feature is because of two factors. First, the proposed LRF is more stable and repeatable compared with the LRFs used in the SHOT, spin image, and RoPS features (see Figs. 3 and 9). Second, the TriSI feature represents a more discriminative description of the local surface compared with the spin image, SHOT, and RoPS features. Note that although the same LRF is used by the TriSI and modified spin image features, the proposed TriSI descriptor outperformed the modified spin image in terms of recall by a large margin (see Fig. 9). It is therefore clear that the TriSI feature offers more discriminative power compared to the modified spin image, because the TriSI feature records the shape information of the local surface around the three orthogonal axes rather than just a single axis.

4.3. Performance on the U3OR Dataset

We further tested the performance of our TriSI feature on the U3OR dataset. The U3OR dataset is one of the most widely used real-life datasets in 3D computer vision [30, 1, 32, 40]. It contains five models (namely, Chef, Chicken, Parasaurolophus, T-Rex, and Rhino) and 50 real scenes [30]. Each scene contains four to five objects in the presence of clutter and occlusion and was acquired with a Minolta Vivid 910 scanner. Each model was reconstructed from several point clouds which were scanned at different viewpoints. Two sample models and two sample scenes are shown in Fig. 10. One thousand ground-truth point correspondences were generated from each scene and its corresponding models. Different features were then extracted at these feature points. TriSI features were further compressed using the PCA subspace trained on the model features. The lengths for the compressed TriSI, spin image, SHOT, and RoPS features were 48, 225, 320, and 135, respectively. These features were tested in terms of RPC with respect to occlusion and clutter of the objects present in the scenes (refer to [30, 17] for the definitions of occlusion and clutter).



(a) T-Rex (b) Rhino (c) Scene 1 (d) Scene 2

Figure 10: Two sample models and two sample scenes of the U3OR dataset (Figure best seen in color).

4.3.1. Robustness to Occlusion

The RPC results with respect to different levels of occlusion of the underlying objects are shown in Fig. 11. Several observations can be made from these results. First, the TriSI feature achieved the best performance under all levels of occlusion, followed by RoPS and spin image. Second, the TriSI feature obtained nearly the same performance under low and medium levels of occlusion (as shown in Figs. 11(a) and (b)). Then, its performance decreased slightly under a high level of occlusion (as shown in Fig. 11(c)). This clearly indicates that the TriSI feature is very robust to occlusion. The robustness is because that a TriSI feature is generated from a small patch of the scene. Therefore, the information in the local surface is not sensitive to the missing points of the whole shape (because of occlusion). The slight performance drop in Fig. 11(c) is because of an excessive level of occlusion. That is, for the scenes with a high level of occlusion, the missing points significantly affect the completeness of the local surface, and ultimately the descriptiveness of the extracted local feature.

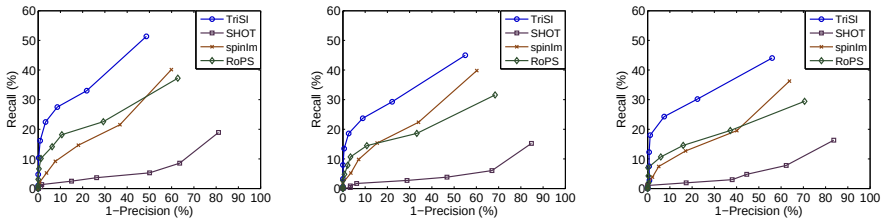


Figure 11: Recall vs 1-precision curves with respect to occlusion. (a) Occlusion between 60% to 70%. (b) Occlusion between 70% to 80%. (c) Occlusion between 80% to 90%.

4.3.2. Robustness to Clutter

The RPC results with respect to different levels of clutter are shown in Fig. 12. It can be seen that i) the TriSI feature achieved the best performance compared to the RoPS, SHOT, and spin image features. ii) the TriSI feature achieved similar results under low and medium levels of clutter and a relatively low performance under a high level of clutter. These conclusions are similar to those made with respect to occlusion (Section 4.3.1). Note that the robustness to occlusion and clutter is one of the major advantages of a local surface feature compared with its global counterparts.

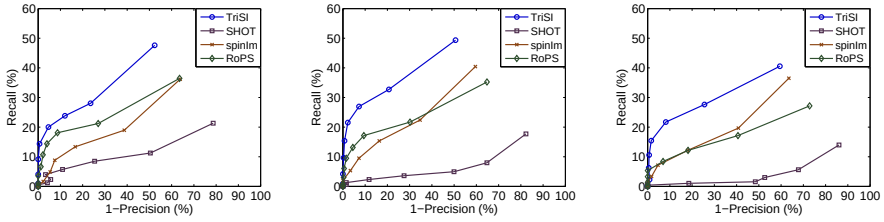


Figure 12: Recall vs 1-precision curves with respect to clutter. (a) Clutter between 40% to 60%. (b) Clutter between 60% to 80%. (c) Clutter between 80% to 100%.

5. Evaluation of the 3D Object Recognition Algorithm

To test the performance of our proposed 3D object recognition algorithm, we conducted extensive object recognition experiments on four standard datasets (i.e., BoD1 [42], U3OR [30], QuLD [40], and CFVD [35]). These datasets incorporate several variations including complicated background, various poses, real noise, varying mesh resolutions, occlusion, clutter, and different imaging techniques. We used the compressed TriSI feature with our object recognition algorithm to perform experiments on these datasets. We also present the results, which were originally reported by the state-of-the-art algorithms in the literature.

5.1. Results on BoD1 Dataset

We first added different levels of Gaussian noise with standard deviations of 0.1 mr, 0.2 mr, 0.3 mr, 0.4 mr, and 0.5 mr to the 45 scenes, and the recognition rates are shown in Fig. 13(a). We then added different levels of shot noise with outlier ratios of 0.2%, 0.5%, 1.0%, 2.0%, and 5.0% to the 45 scenes (the same as in [48]), and the recognition rates are shown in Fig. 13(b). We further resampled the noise-free scenes to 1/2, 1/4, 1/8, and 1/16 of their original mesh resolution, and the recognition rates are presented in Fig. 13(c).

5.1.1. Results under Different Types of Noise

Figure 13(a) shows that the TriSI-, SHOT- and RoPS-based algorithms achieved a high recognition rate of 100% at all levels of Gaussian noise. The recognition rate of the spin image-based algorithm was 100% under minor Gaussian noise with a standard deviation of 0.1 mr. However, it dropped significantly when the standard deviation of the Gaussian noise was larger than 0.2 mr. These recognition results are consistent with the feature matching results shown in Fig. 6. Because the matching accuracy of the spin image decreased as the level of noise increased, it was reasonable to expect that the recognition performance deteriorated.

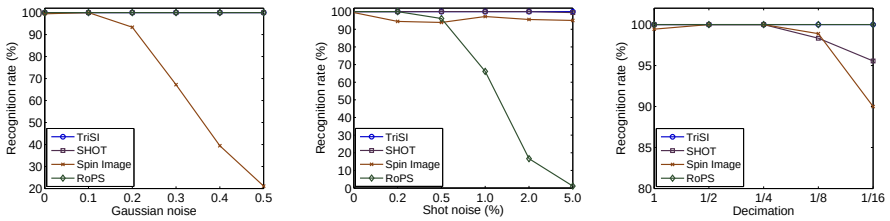


Figure 13: Recognition rates on the BoD1 dataset. (a) Gaussian noise. (b) Shot noise. (c) Varying mesh resolutions.

As shown in Fig. 13(b), the TriSI-based algorithm achieved a high recognition rate of 100% at all levels of shot noise. The SHOT-based algorithm achieved the closest performance. It achieved a recognition rate of 100% with outlier ratios for shot noise of no more than 2.0%. As the outlier ratio of shot noise increased to 5.0%, the performance of the SHOT-based algorithm declined slightly. The spin image-based algorithm achieved a relatively poor performance at all levels of shot noise. The RoPS-based algorithm was very sensitive to shot noise. It achieved an acceptable recognition rate when the outlier ratio for shot noise is less than 0.5%. Its performance however deteriorated significantly as shot noise increased.

5.1.2. Results under Varying Mesh Resolutions

Fig. 13(c) indicates that the TriSI-based algorithm achieved a 100% recognition rate at all levels of mesh resolution. The recognition rate of the SHOT-based

algorithm was 100% when the mesh decimation was larger than 1/8, and then it decreased to 99.4% under 1/16 mesh decimation. The spin image-based algorithm was substantially more sensitive to varying mesh resolutions compared to the TriSI- and SHOT-based algorithms. Its recognition rate deteriorated sharply when the number of vertices in the decimated scenes was less than 1/8 of their original number.

Overall, our proposed TriSI-based algorithm achieved the best performance with respect to both noise and varying mesh resolutions. In contrast, the spin image-based algorithm was very sensitive to noise and varying mesh resolutions.

5.1.3. Computational Complexity

To evaluate the computational complexity, we conducted timing experiments for the whole 3D object recognition pipeline on the BOD1 dataset using compressed TriSI, spin image, and SHOT features. These algorithms were implemented in nonoptimized MATLAB codes without using any parallel programming. The experiments were run on an Intel Core i7-2700K 3.5GHz windows machine with 16GB RAM. Because the runtime is related to the number of points in a scene, we resampled the scenes down to 1/2, 1/4, 1/8, and 1/16 of their original mesh resolution, and recorded the runtime at each mesh resolution level. The average number of points for the models and original scenes are 45,998 and 191,649, respectively. The average runtime results for recognizing an object instance in each scene are summarized in Table 1.

Table 1: Average online runtime with respect to varying mesh resolutions on the BoD1 dataset. Here, "SI" stands for "Spin Image".

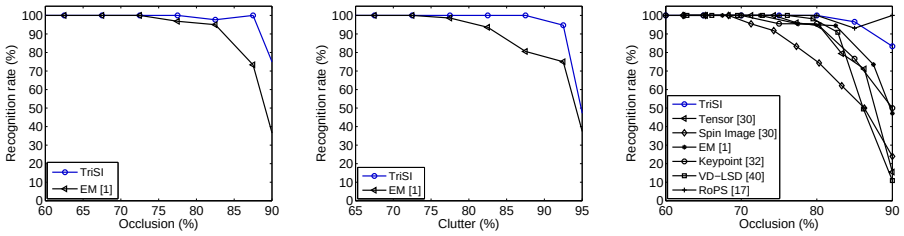
Rate	#Points	Feature Generation (s)			Object Recognition (s)			Overall (s)		
		TriSI	SHOT	SI	TriSI	SHOT	SI	TriSI	SHOT	SI
1	191649	40.3	14.4	14.0	43.6	43.7	45.2	83.9	58.1	59.2
1/2	95652	19.7	7.1	6.3	26.5	32.7	27.2	46.2	39.8	33.5
1/4	48120	10.0	3.6	3.2	19.7	22.3	36.7	29.7	25.9	40.0
1/8	24232	4.8	1.8	1.6	12.5	21.5	36.5	17.2	23.3	38.2
1/16	12153	2.1	1.0	0.8	12.2	34.7	118.8	14.3	35.8	119.6

It was observed that i) the average overall runtime increased with the number of points in a scene. For scenes with less than 48120 points, the TriSI-based algorithm consumed less computational time compared to the SHOT- and spin image-based algorithms. As the average number of scene points increased, the TriSI-based algorithm cost relatively more overall time for the whole 3D object recognition pipeline. ii) The generation of TriSI features consumed more computational time compared to the generation of the SHOT and Spin image features. However, the TriSI-based algorithm was more efficient during the object recognition phase because the matching results of the TriSI features are more accurate and reliable compared to the SHOT and spin image features (as shown in Section 4). Therefore, the hypothesis verification module (as shown in Section 3.2) of the object recognition pipeline can be performed more effi-

ciently. Note that the computation can further be accelerated by employing several speedup strategies, such as parallel programming and C or GPU implementations.

5.2. Results on the U3OR Dataset

The U3OR dataset contains five models and 50 real scenes. More details regarding the dataset are presented in Section 4.3. To achieve a fair comparison with [1], Figs. 14 (a) and (b) show the recognition rates for the five models on the 50 scenes with respect to occlusion and clutter, respectively. The results of the EM-based algorithm (which was reported in [1]) are also plotted in Figs. 14 (a) and (b). The TriSI-based algorithm outperformed the EM-based algorithm [1], especially in the presence of significant occlusion and clutter. The average recognition rate of TriSI-based algorithm was 99.1%, which was higher than the average recognition rate of 93.6% achieved by the EM-based algorithm.



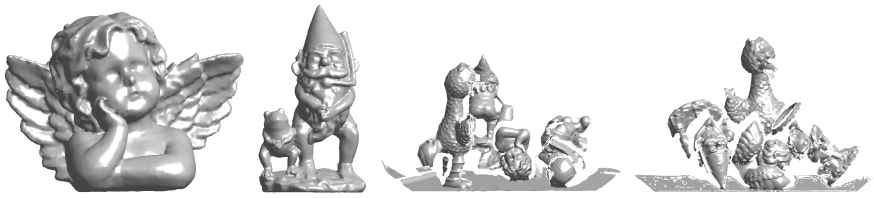
(a) (b) (c)

Figure 14: Recognition rates on the U3OR dataset. (a) Recognition rates for five models with respect to occlusion. (b) Recognition rates for five models with respect to clutter. (c) Recognition rates for four models with respect to occlusion.

To make a fair comparison with [30, 1, 32, 40, 17], Fig. 14 (c) presents the recognition rates for four models on the 50 scenes with respect to occlusion. We used the same dataset and experimental setup as [30, 1, 32, 40, 17] and excluded the model Rhino from our recognition results. Our TriSI-based algorithm achieved the best recognition results, and it obtained an average recognition rate of 99.4% with up to 84% occlusion. In contrast, the recognition rates of EM- [1], tensor- [30], spin image- [30], and RoPS- [17] based algorithms with up to 84% occlusion were 97.5%, 96.6%, 87.8%, and 98.8%, respectively. Moreover, the TriSI-based algorithm was robust to severe occlusions. It achieved a recognition rate of more than 80% even under 90% occlusion.

5.3. Results on the QuLD Dataset

The QuLD dataset contains five models and 80 scenes [40]. Each scene was acquired with an LIDAR sensor and contains one, three, four or five objects in the presence of clutter and occlusion. Each model was generated by merging several point clouds. Fig. 15 shows two sample models and two sample scenes.



(a) Angle (b) Gnome (c) Scene 1 (d) Scene 2

Figure 15: Two sample models and two sample scenes of the QuLD dataset (Figure best seen in color).

To make a rigorous comparison with the results given by [1, 17], we first performed 3D object recognition experiments on the full dataset, which contains 80 scenes. The recognition rates are presented in Table 2 with a comparison of the results reported in [1, 17]. The average recognition rate of the TriSI-based algorithm was higher than the EM-based algorithm [1] by a margin of 14.7%. The proposed TriSI-based algorithm also outperformed our previous work (i.e., RoPS-based algorithm) [17] with average recognition rates of 97.1% and 95.4%, respectively. Moreover, the proposed TriSI-based algorithm achieved the best results for all individual models.

Table 2: Recognition rates on the full QuLD dataset. The best results are in bold faces.

Algorithm	Angel (%)	Big-bird (%)	Gnome (%)	Kid (%)	Zoe (%)	Average (%)
TriSI	**100**	**100**	**100**	**97.9**	**87.5**	**97.1**
RoPS [17]	97.9	100	97.9	95.8	85.4	95.4
EM [1]	77.1	87.5	87.5	83.3	76.6	82.4

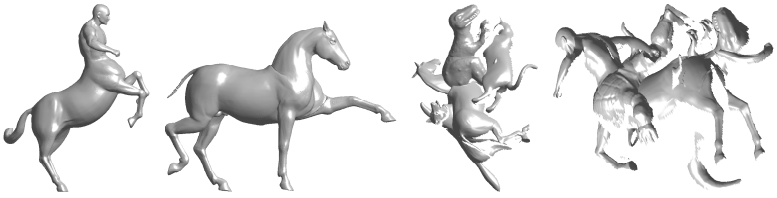
To make a direct comparison with the results given by [1, 40, 17], we then performed 3D object recognition on the same subset dataset (which contains 55 scenes) as in [1, 40, 17]. The recognition rates on the subset dataset are presented in Table 3 with a comparison to the results achieved by EM- [1], VD-LSD-(SQ) [40], VD-LSD- (VQ) [40], 3DSC- [10, 40], spin image- [23, 40], spin image spherical- [22, 40], and RoPS- [17] based algorithms. Our average recognition rate in this case was 96.9%, which was higher than the best result reported in the literature by RoPS [17]. The TriSI-based algorithm achieved a 100% recognition rate for models Angel, Big-bird, and Gnome. It also achieved the highest recognition rates for both models Kid and Zoe.

5.4. Results on the CFVD Dataset

The CFVD dataset contains 20 models and 150 scenes [35]. It is the most challenging and is currently the largest publicly available dataset for 3D object recognition [35]. Each scene was captured with a virtual camera and contains three to five objects. Fig. 16 shows two sample models and two sample scenes.

Table 3: Recognition rates on a subset of the QuLD dataset. 'NA' indicates that the corresponding item is not available. The best results are in bold faces.

Algorithm	Angel (%)	Big-bird (%)	Gnome (%)	Kid (%)	Zoe (%)	Average (%)
TriSI	**100**	**100**	**100**	**97.4**	**87.2**	**96.9**
RoPS [17]	97.4	100	97.4	94.9	87.2	95.4
EM [1]	NA	NA	NA	NA	NA	81.9
VD-LSD (SQ) [40]	89.7	100	70.5	84.6	71.8	83.8
VD-LSD (VQ) [40]	56.4	97.4	69.2	51.3	64.1	67.7
3DSC [40]	53.8	84.6	61.5	53.8	56.4	62.1
Spin Image [40]	53.8	84.6	38.5	51.3	41.0	53.8
Spin Image Spherical [40]	53.8	74.4	38.5	61.5	43.6	54.4



(a) Centaur1 (b) Horse7 (c) Scene 1 (d) Scene 2

Figure 16: Two sample models and two sample scenes of the CFVD dataset (Figure best seen in color).

As in [35], we left two models out of the 20 models from recognition tests and used these two models to produce additional clutter. Table 4 shows the recognition results on the dataset. We present the recall and precision results for each individual object together with the average results on all of the 18 objects. For a direct comparison, we also present the results reported in [17] and [35] (which used SHOT features and a game theory (SHOT+Game) for object recognition). Our average recall was 96.7%, which was better than the recall of 94.7% reported in [35]. The TriSI-based algorithm achieved the best recall performance on 15 individual objects out of a total of 18 objects. The average precision of the TriSI-based algorithm was 99.3%, which outperformed the result achieved by the SHOT+Game algorithm (reported in [35]) by a margin of 6.3%. Note that the TriSI-based algorithm also achieved better recall and precision results compared to the RoPS-based algorithm [17]. The TriSI-based algorithm obtained the best precision results on 16 individual objects. That is, the TriSI-based algorithm results in less false positives compared with the other algorithms.

Table 4: Recognition results on the CFVD dataset. (a) Recall (%). (b) Precision (%). The best results are in bold faces.

(a) Recall results.

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
TriSI	**100**	**100**	60	**100**	**100**	**100**	**91**	**100**	**100**	96
SHOT+Game [35]	97	97	82	100	100	100	86	89	**95**	100
RoPS [17]	100	100	44	100	100	100	91	100	100	100

	M11	M12	M13	M14	M15	M16	M17	M18	Average
TriSI	**100**	**100**	96	**100**	**100**	**100**	**95**	**100**	96.7
RoPS [17]	100	100	100	97	100	100	95	100	96.0
SHOT+Game [35]	91	100	100	94	91	97	83	95	94.7

(b) Precision results.

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
TriSI	**100**	**100**	**100**	**100**	93	**100**	**100**	**100**	**100**	**100**
RoPS [17]	97	100	100	100	100	97	100	100	100	100
SHOT+Game [35]	100	100	78	96	93	93	95	100	91	89

	M11	M12	M13	M14	M15	M16	M17	M18	Average
TriSI	**100**	97	**100**	**97**	**100**	**100**	**100**	**100**	**99.3**
RoPS [17]	100	100	100	97	96	100	100	100	99.1
SHOT+Game [35]	95	97	88	97	91	97	83	82	93.0

6. Summary and Discussion

Based on the experimental results and analysis in Sections 4 and 5, it can be summarized that i) the compressed TriSI feature outperforms existing feature descriptors including the spin image and SHOT features on both the BoD1 and U3OR datasets. The TriSI feature is very robust to Gaussian noise, Laplacian noise, shot noise, varying mesh resolutions, occlusion, and clutter. It is also very compact because its length is shorter compared to the spin image, SHOT, and RoPS features. Therefore, The compressed TriSI feature is not only rich in information content, but also compact in dimensionality. ii) The TriSI-based algorithm achieves the best overall object recognition results on four standard datasets. Note that the datasets were acquired with different techniques, including the synthetic simulation, LIDAR, and triangulation laser scanner (e.g., Minolta Vivid 910). These results clearly demonstrate the superiority of our proposed TriSI-based algorithm.

Although the proposed TriSI feature is only used for 3D object recognition in this paper, other possible application areas include 3D model/shape retrieval [2], 3D face recognition [47, 26, 27], and 3D object/scene reconstruction [16, 21]. For example, the idea to construct an LRF using all of the implicit information of a local surface can also be extended to improve the accuracy for pose normalization in 3D model retrieval and 3D face recognition. The TriSI features

can further be integrated with different classification algorithms to perform 3D shape retrieval. It is also possible to improve the performance of existing 3D reconstruction algorithms using the proposed highly descriptive and robust TriSI feature.

In spite of these positive points, our TriSI feature is not without limitation. The TriSI feature is designed for rigid objects, and it may face challenges when dealing with deformable objects. In future works, the intrinsic geodesics can be integrated with the TriSI feature to achieve invariance to isometric deformations. In addition, the TriSI feature is best suited for objects with rich shape and geometric information. Completely symmetric and bland objects (e.g., a balloon) or planar objects (e.g., a wall) cannot be addressed and are out of the scope of these types of features. However, these objects can be addressed by integrating both shape and color information [18]. Finally, all object recognition experiments were conducted in indoor environments. In future works, we intend to explore in depth the application of the TriSI-based algorithm for large-scale object recognition from 3D point clouds in urban environments.

7. Conclusion

In this paper, we proposed a novel TriSI feature for local surface description. Feature matching experiments demonstrated that the TriSI feature was highly descriptive. It was also very robust to Gaussian noise, Laplacian noise, shot noise, varying mesh resolutions, occlusion, and clutter. We also proposed a hierarchical 3D object recognition algorithm. Extensive experiments were performed on a number of standard and challenging datasets that incorporate a set of variations including complicated backgrounds, real noise, varying mesh resolutions, occlusion, clutter, and various imaging techniques. The Experimental results showed that the TriSI-based algorithm achieved the best overall results on all of these datasets. It consistently outperformed the state-of-the-art algorithms. Our future work will include an implementation of the proposed TriSI feature in C++ that is compatible with the Point Cloud Library (PCL) [37].

Acknowledgments

This research is supported by a China Scholarship Council (CSC) scholarship, a National Natural Science Foundation of China (NSFC) grant (2011611067), and two Australian Research Council grants (DE120102960, DP110102166).

References

[1] P. Bariya, J. Novatnack, G. Schwartz, and K. Nishino. 3D geometric scale variability in range images: Features and descriptors. *International Journal of Computer Vision*, 99(2):232–255, 2012.

[2] B. Bustos, D.A. Keim, D. Saupe, T. Schreck, and D.V. Vranić. Feature-based similarity search in 3D object databases. *ACM Computing Surveys*, 37(4):345–387, 2005.

[3] E. J. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006.

[4] C. S. Chua and R. Jarvis. Point signatures: A new representation for 3D object recognition. *International Journal of Computer Vision*, 25(1):63–85, 1997.

[5] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *23rd Annual Conference on Computer Graphics and Interactive Techniques*, pages 303–312, 1996.

[6] T. Darom and Y. Keller. Scale invariant features for 3D mesh models. *IEEE Transactions on Image Processing*, 21(5):2758 – 2769, 2012.

[7] F. De la Torre and M. J. Black. Robust principal component analysis for computer vision. In *8th IEEE International Conference on Computer Vision*, volume 1, pages 362–369, 2001.

[8] H.Q. Dinh and S. Kropac. Multi-resolution spin-images. In *IEEE International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 863–870, 2006.

[9] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[10] A. Frome, D. Huber, R. Kolluri, T. Bülow, and J. Malik. Recognizing objects in range data using regional point descriptors. In *8th European Conference on Computer Vision*, pages 224–237, 2004.

[11] Y. Gao, Q. Dai, and N.-Y. Zhang. 3D model comparison using spatial structure circular descriptor. *Pattern Recognition*, 43(3):1142–1151, 2010.

[12] Y. Gao, M. Wang, Z.-J. Zha, Q. Tian, Q. Dai, and N. Zhang. Less is more: efficient 3-D object retrieval with query view selection. *IEEE Transactions on Multimedia*, 13(5):1007–1018, 2011.

[13] Y. Gao, J. Tang, R. Hong, S. Yan, Q. Dai, N. Zhang, and T.-S. Chua. Camera constraint-free view-based 3-D object retrieval. *IEEE Transactions on Image Processing*, 21(4):2269–2281, 2012.

[14] Y. Gao, M. Wang, R. Ji, Z. Zha, and J. Shen. k-partite graph reinforcement and its application in multimedia information retrieval. *Information Sciences*, 194:224–239, 2012.

[15] Y. Gao, M. Wang, D. Tao, R. Ji, and Q. Dai. 3-D object retrieval and recognition with hypergraph analysis. *IEEE Transactions on Image Processing*, 21(9):4290–4303, 2012.

[16] Y. Guo, F. Sohel, M. Bennamoun, M. Lu, and J. Wan. TriSI: A distinctive local surface descriptor for 3D modeling and object recognition. In *8th International Conference on Computer Graphics Theory and Applications*, pages 86–93, 2013.

[17] Y. Guo, F. Sohel, M. Bennamoun, M. Lu, and J. Wan. Rotational projection statistics for 3D local surface description and object recognition. *International Journal of Computer Vision*, 105(1):63–86, 2013.

[18] Y. Guo, F. Sohel, M. Bennamoun, J. Wan, and M. Lu. Integrating shape and color cues for textured 3D object recognition. In *The 8th IEEE Conference on Industrial Electronics and Applications*, 2013.

[19] Y. Guo, J. Wan, M. Lu, and W. Niu. A parts-based method for articulated target recognition in laser radar data. *Optik - International Journal for Light and Electron Optics*, 124(17):2727–2733, 2013.

[20] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, and J. Wan. 3D object recognition in cluttered scenes with local surface features: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence, in press*, 2014.

[21] Y. Guo, F. Sohel, M. Bennamoun, J. Wan, and M. Lu. An accurate and robust range image registration algorithm for 3D object modeling. *IEEE Transactions on Multimedia*, 16(5):1377–1390, 2014.

[22] A. E. Johnson. Surface landmark selection and matching in natural terrain. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 413–420, 2000.

[23] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):433–449, 1999.

[24] U. Kang, M. Hebert, and S. Park. Fast and scalable approximate spectral graph matching for correspondence problems. *Information Sciences*, 220: 306–318, 2013.

[25] I. Kokkinos, M.M. Bronstein, R. Litman, and A.M. Bronstein. Intrinsic shape context descriptors for deformable shapes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 159–166, 2012.

[26] Y. Lei, M. Bennamoun, and A.A. El-Sallam. An efficient 3D face recognition approach based on the fusion of novel local low-level features. *Pattern Recognition*, 46(1):24–37, 2013.

[27] Y. Lei, M. Bennamoun, M. Hayat, and Y. Guo. An efficient 3D face recognition approach using local geometrical signatures. *Pattern Recognition*, 47 (2):509–524, 2014.

[28] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[29] G.-F. Lu and Y. Wang. Feature extraction using a fast null space based linear discriminant analysis algorithm. *Information Sciences*, 193:72–80, 2012.

[30] A.S. Mian, M. Bennamoun, and R. Owens. Three-dimensional model-based object recognition and segmentation in cluttered scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1584–1601, 2006.

[31] A.S. Mian, M. Bennamoun, and R. A. Owens. A novel representation and feature matching algorithm for automatic pairwise registration of range images. *International Journal of Computer Vision*, 66(1):19–40, 2006.

[32] A.S. Mian, M. Bennamoun, and R. Owens. On the repeatability and quality of keypoints for local feature-based 3D object retrieval from cluttered scenes. *International Journal of Computer Vision*, 89(2):348–361, 2010.

[33] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27 (10):1615–1630, 2005.

[34] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Shape distributions. *ACM Transactions on Graphics*, 21(4):807–832, 2002.

[35] E. Rodolà, A. Albarelli, F. Bergamasco, and A. Torsello. A scale independent selection process for 3D object recognition in cluttered scenes. *International Journal of Computer Vision*, pages 1–17, 2013.

[36] S. Ruiz-Correa, L.G. Shapiro, and M. Melia. A new signature-based method for efficient 3-D object recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages I–769, 2001.

[37] R. B. Rusu and S. Cousins. 3D is here: Point cloud library (PCL). In *IEEE International Conference on Robotics and Automation*, pages 1–4, 2011.

[38] L. Shang and M. Greenspan. Real-time object recognition in sparse range images using error surface embedding. *International Journal of Computer Vision*, 89(2):211–228, 2010.

[39] F. M. Sukno, J. L. Waddington, and P. F. Whelan. Rotationally invariant 3D shape contexts using asymmetry patterns. In *8th International Conference on Computer Graphics Theory and Applications*, 2013.

[40] B. Taati and M. Greenspan. Local shape descriptor selection for object recognition in range data. *Computer Vision and Image Understanding*, 115(5):681–694, 2011.

[41] B. Taati, M. Bondy, P. Jasiobedzki, and M. Greenspan. Variable dimensional local shape descriptors for object recognition in range data. In *11th IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[42] F. Tombari, S. Salti, and L. Di Stefano. Unique signatures of histograms for local surface description. In *European Conference on Computer Vision*, pages 356–369. Springer, 2010.

[43] F. Tombari, S. Salti, and L. Di Stefano. Unique shape context for 3D data description. In *ACM Workshop on 3D Object Retrieval*, pages 57–62, 2010.

[44] F. Tombari, S. Salti, and L. Di Stefano. Performance evaluation of 3D keypoint detectors. *International Journal of Computer Vision*, 102(1):198–220, 2013.

[45] J. Wang, Y. Lu, L. Gu, C. Zhou, and X. Chai. Moving objects recognition under simulated prosthetic vision using background-subtraction-based image processing strategies. *Information Sciences*, 2014.

[46] H. Xu, C. Caramanis, and S. Sanghavi. Robust PCA via outlier pursuit. *IEEE Transactions on Information Theory*, 58(5):3047–3064, 2012.

[47] Y. Xu, Q. Zhu, Z. Fan, D. Zhang, J. Mi, and Z. Lai. Using the idea of the sparse representation to perform coarse-to-fine face recognition. *Information Sciences*, 238:138–148, 2013.

[48] A. Zaharescu, E. Boyer, and R. Horaud. Keypoints and local descriptors of scalar functions on 2D manifolds. *International Journal of Computer Vision*, 100:78–98, 2012.

[49] X. Zhao, C. Zhang, L. Xu, B. Yang, and Z. Feng. IGA-based point cloud fitting using B-spline surfaces for reverse engineering. *Information Sciences*, 2013.

[50] Y. Zhong. Intrinsic shape signatures: A shape descriptor for 3D object recognition. In *IEEE International Conference on Computer Vision Workshops*, pages 689–696, 2009.