TriSI: A Distinctive Local Surface Descriptor for 3D Modeling and Object Recognition

Yulan Guo[1,2], Ferdous Sohel[2], Mohammed Bennamoun[2], Min Lu[1] and Jianwei Wan[1]
[1]*College of Electronic Science and Engineering, National University of Defense Technology, Changsha, P.R.China*
[2]*School of Computer Science and Software Engineering, The University of Western Australia, Perth, Australia*
yulan.guo@nudt.edu.cn, {ferdous.sohel, mohammed.bennamoun}@uwa.edu.au, {lumin, kermitwjw}@nudt.edu.cn

Keywords: Local Surface Descriptor, 3D Modeling, 3D Object Recognition, Range Image Registration.

Abstract: Local surface description is a critical stage for surface matching. This paper presents a highly distinctive local surface descriptor, namely TriSI. From a keypoint, we first construct a unique and repeatable local reference frame (LRF) using all the points lying on the local surface. We then generate three spin images from the three coordinate axes of the LRF. These spin images are concatenated and further compressed into a TriSI descriptor using the principal component analysis technique. We tested our TriSI descriptor on the Bologna Dataset and compared it to several existing methods. Experimental results show that TriSI outperformed existing methods under all levels of noise and varying mesh resolutions. The TriSI was further tested to demonstrate its effectiveness in 3D modeling. Experimental results show that it can accurately perform pairwise and multiview range image registration. We finally used the TriSI descriptor for 3D object recognition. The results on the UWA Dataset show that TriSI outperformed the state-of-the-art methods including spin image, tensor and exponential map. The TriSI based method achieved a high recognition rate of 98.4%.

1 INTRODUCTION

Surface matching is a fundamental research topic in both 3D Computer Vision and Computer Graphics. It has a number of applications, including 3D modeling (Mian et al., 2006), 3D shape retrieval (Shilane et al., 2004), 3D object recognition (Attene et al., 2011; Guo et al., 2012; Guo et al., 2013), 3D mapping (Huber et al., 2000), robotics (Lai et al., 2011b), and reverse engineering(Williams and Bennamoun, 2000). With the rapid development of low-cost 3D scanners (e.g., Microsoft Kinect), range images are becoming more available (Lai et al., 2011a; Rusu and Cousins, 2011). The data availability together with the progress in high-speed computing devices have increased the demand for efficient and accurate range image representation techniques (Mian et al., 2010).

There are two basic approaches to represent a range image, namely global feature and local feature based approaches (Salti et al., 2011). A global feature based approach uses a global feature to represent a surface. It is very popular in 3D shape retrieval, but it is sensitive to occlusion and clutter. A local feature based approach however, uses a set of 3D keypoints and local surface descriptors to represent a surface. It is therefore, suitable to surface matching in the presence of occlusion and clutter (Salti et al., 2011). In the process of surface matching, the distinctiveness and robustness of the local surface descriptors play a significant role (Taati and Greenspan, 2011).

A number of papers on local surface descriptors can be found in the literature (Bustos et al., 2005; Bronstein et al., 2010; Boyer et al., 2011). Chua and Jarvis (1997) proposed a Point Signature by recording the signed distances between the neighboring surface points and their correspondences in a fitted plane. Point Signature is however, sensitive to noise and varying mesh resolutions (Mian et al., 2005). Johnson and Hebert (1999) proposed a Spin Image descriptor by accumulating the neighboring points into a 2D histogram. The spin image is one of the most cited methods in the literature. It is however weakly distinctive and sensitive to varying mesh resolutions (Mian et al., 2010). Chen and Bhanu (2007) used Local Surface Patches (LSP) to represent a range image. Since the LSP descriptor requires the calculation of second-order derivatives of a surface, it is sensitive to noise. Flint et al. (2007) introduced the THRIFT descriptor by generating a 1D histogram according to the surface normal deviations. Tombari et al. (2010) proposed a Signature of Histograms of OrienTations (SHOT) by encoding the surface normal deviations in a parti-

tioned spherical neighborhood. The SHOT is highly descriptive. It is however, also sensitive to varying mesh resolutions. Other descriptors include Point's Fingerprint (Sun and Abidi, 2001), 3D Shape Context (Frome et al., 2004), Tensor (Mian et al., 2006), Exponential Map (EM) (Novatnack and Nishino, 2008) and Variable-Dimensional Local Shape Descriptors (VD-LSD) (Taati and Greenspan, 2011).

Most of the existing local surface descriptors suffer from low descriptiveness, or robustness to noise, or sensitivity to varying mesh resolutions (Bariya et al., 2012). Motivated by these limitations, we propose a highly distinctive and robust local surface descriptor called Tri-Spin-Image (TriSI). The TriSI is an improvement of the Spin Image descriptor. It first builds a unique and repeatable Local Reference Frame (LRF) for each keypoint. It then generates three spin images by spinning one sheet around each axis of the LRF. The three images are concatenated to form an overall TriSI descriptor. The TriSI descriptor is further compressed using the Principal Component Analysis (PCA) technique. Performance evaluation results show that our proposed TriSI descriptor is highly descriptive. It is very robust to both noise and varying mesh resolutions. The effectiveness of TriSI descriptor was also demonstrated by 3D modeling including pairwise and multiview range image registration. The TriSI descriptor was further used for 3D object recognition and was tested on the UWA Dataset. Experimental results show that TriSI outperformed the state-of-the-art methods including Spin Image, Tensor, EM and VD-LSD.

The rest of this paper is organized as follows: Section 2 describes the TriSI surface descriptor. Section 3 presents the feature matching performance. Section 4 demonstrates the TriSI based 3D modeling methods and their experimental results. Section 5 presents the 3D object recognition results. Section 6 concludes this paper.

2 TriSI SURFACE DESCRIPTOR

The process of generating a TriSI surface descriptor includes three modules, i.e., LRF construction, TriSI generation and TriSI compression.

2.1 LRF Construction

Given a triangular mesh surface $\mathcal{S}$ and a set of keypoints $\{\boldsymbol{o}_1, \boldsymbol{o}_2, \cdots, \boldsymbol{o}_K\}$, a set of local surface descriptors $\{\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_K\}$ should be generated to represent the surface $\mathcal{S}$. Here, K denotes the number of keypoints on the surface $\mathcal{S}$. For a given keypoint $\boldsymbol{o}_k, k = 1, 2, \cdots, K$, we first extract the local surface $\mathcal{L}$ using a sphere of radius r centered at $\boldsymbol{o}_k$. We then construct a LRF for $\boldsymbol{o}_k$ using all the points lying on the local surface $\mathcal{L}$ rather than using just the mesh vertices.

Assume that the local surface $\mathcal{L}$ contains N triangles and M vertices. For the ith triangle with vertices $\boldsymbol{q}_{i1}$, $\boldsymbol{q}_{i2}$ and $\boldsymbol{q}_{i3}$, we calculate the scatter matrix $\mathbf{S}_i$ using the continuous PCA algorithm:

$$\begin{aligned}\mathbf{S}_i &= \frac{1}{12}\sum_{j=1}^{3}\sum_{n=1}^{3}\left(\boldsymbol{q}_{ij}-\boldsymbol{o}_k\right)\left(\boldsymbol{q}_{in}-\boldsymbol{o}_k\right)^{\mathrm{T}} \\ &\quad +\frac{1}{12}\sum_{j=1}^{3}\left(\boldsymbol{q}_{ij}-\boldsymbol{o}_k\right)\left(\boldsymbol{q}_{ij}-\boldsymbol{o}_k\right)^{\mathrm{T}}. \qquad (1)\end{aligned}$$

The overall scatter matrix $\mathbf{S}$ of the local surface $\mathcal{L}$ is then calculated as:

$$\mathbf{S}=\sum_{i=1}^{N}\gamma_{i1}\gamma_{i2}\mathbf{S}_i, \qquad (2)$$

where the weights γ_{i1} and γ_{i2} are respectively defined as:

$$\gamma_{i1}=\frac{\left|\left(\boldsymbol{q}_{i2}-\boldsymbol{q}_{i1}\right)\times\left(\boldsymbol{q}_{i3}-\boldsymbol{q}_{i1}\right)\right|}{\sum_{i=1}^{N}\left|\left(\boldsymbol{q}_{i2}-\boldsymbol{q}_{i1}\right)\times\left(\boldsymbol{q}_{i3}-\boldsymbol{q}_{i1}\right)\right|}, \qquad (3)$$

$$\gamma_{i2}=\left(r-\left|\boldsymbol{o}_k-\frac{\boldsymbol{q}_{i1}+\boldsymbol{q}_{i2}+\boldsymbol{q}_{i3}}{3}\right|\right)^2. \qquad (4)$$

We perform an eigenvalue decomposition on the scatter matrix $\mathbf{S}$ to get three orthogonal eigenvectors $\boldsymbol{v}_1$, $\boldsymbol{v}_2$ and $\boldsymbol{v}_3$. These eigenvectors are in the order of decreasing magnitude of their associated eigenvalues. The eigenvectors $\boldsymbol{v}_1$, $\boldsymbol{v}_2$ and $\boldsymbol{v}_3$ form the basis for the 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 $\mathbf{S}$. We therefore propose a sign disambiguation technique.

We define the unambiguous vectors $\widetilde{\boldsymbol{v}_1}$ and $\widetilde{\boldsymbol{v}_3}$ as:

$$\widetilde{\boldsymbol{v}_1}=\boldsymbol{v}_1\cdot\mathrm{sgn}\left(\sum_{i=1}^{N}\gamma_{i1}\gamma_{i2}\left(\sum_{j=1}^{3}\left(\boldsymbol{q}_{ij}-\boldsymbol{o}_k\right)\boldsymbol{v}_1\right)\right), \qquad (5)$$

$$\widetilde{\boldsymbol{v}_3}=\boldsymbol{v}_3\cdot\mathrm{sgn}\left(\sum_{i=1}^{N}\gamma_{i1}\gamma_{i2}\left(\sum_{j=1}^{3}\left(\boldsymbol{q}_{ij}-\boldsymbol{o}_k\right)\boldsymbol{v}_3\right)\right), \qquad (6)$$

where $\mathrm{sgn}(\cdot)$ denotes the signum function that extracts the sign of a real number. Vector $\widetilde{\boldsymbol{v}_2}$ is then defined as $\widetilde{\boldsymbol{v}_3}\times\widetilde{\boldsymbol{v}_1}$.

Finally, we construct a LRF for the keypoint $\boldsymbol{o}_k$ using $\boldsymbol{o}_k$ as the origin and the unambiguous vectors $\{\widetilde{\boldsymbol{v}_1}, \widetilde{\boldsymbol{v}_2}, \widetilde{\boldsymbol{v}_3}\}$ as the three coordinate axes.

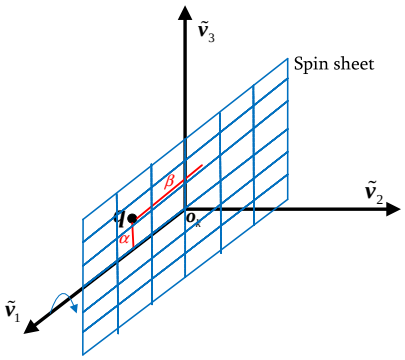


Figure 1: An illustration of the generation of a spin image. (Figure best seen in color.)

2.2 TriSI Generation

Given a keypoint $\boldsymbol{o}_k$, the local surface $\mathcal{L}$ and the LRF vectors $\{\widetilde{\boldsymbol{v}}_1, \widetilde{\boldsymbol{v}}_2, \widetilde{\boldsymbol{v}}_3\}$, we generate three spin images $\{\mathbf{SI}_1, \mathbf{SI}_2, \mathbf{SI}_3\}$ by respectively spinning a sheet about the three axes of the LRF.

We first generate a spin image by spinning a sheet about the $\widetilde{\boldsymbol{v}}_1$ axis. An illustration is shown in Fig. 1. Given the LRF, each point $\boldsymbol{q}$ on the local surface $\mathcal{L}$ is represented by two parameters α and β. Here, α is the perpendicular distance of $\boldsymbol{q}$ from the line which passes through $\boldsymbol{o}_k$ and is parallel to $\widetilde{\boldsymbol{v}}_1$. β is the signed perpendicular distance to the plane which goes through $\boldsymbol{o}_k$ and is perpendicular to $\widetilde{\boldsymbol{v}}_1$, that is:

$$\alpha = \sqrt{\|\boldsymbol{q} - \boldsymbol{o}_k\|^2 - (\widetilde{\boldsymbol{v}}_1 \cdot (\boldsymbol{q} - \boldsymbol{o}_k))^2}, \tag{7}$$

$$\beta = \widetilde{\boldsymbol{v}}_1 \cdot (\boldsymbol{q} - \boldsymbol{o}_k). \tag{8}$$

We accumulate the parameters (α, β) of all the points on $\mathcal{L}$ into a $B \times B$ histogram, where B is the number of bins along each dimension of the histogram. We further bilinearly interpolate this 2D histogram to account for noise. The interpolated histogram is our improved spin image $\mathbf{SI}_1$.

We then generate two other spin images $\mathbf{SI}_2$ and $\mathbf{SI}_3$ by following the exact same way we adopted to generate $\mathbf{SI}_1$. That is, $\mathbf{SI}_2$ and $\mathbf{SI}_3$ are generated by substituting the $\widetilde{\boldsymbol{v}}_1$ in Equations (7-8) with $\widetilde{\boldsymbol{v}}_2$ and $\widetilde{\boldsymbol{v}}_3$, respectively. We finally concatenate the three spin images to obtain our TriSI descriptor, that is:

$$\mathbf{TriSI} = \{\mathbf{SI}_1, \mathbf{SI}_2, \mathbf{SI}_3\}. \tag{9}$$

2.3 TriSI Compression

The dimensionality of the TriSI descriptor is $3B^2$, and it is therefore too large for an efficient feature matching. For example, if B is set to be 15, the dimensionality of a TriSI is 675. We therefore project the TriSI to a PCA subspace to get a more compact feature descriptor. The PCA subspace can be learned from a set of training descriptors $\{\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_T\}$, where T is the number of training descriptors. We calculate the scatter matrix $\mathbf{M}$ as:

$$\mathbf{M} = \sum_{i=1}^{T} (\boldsymbol{f}_i - \overline{\boldsymbol{f}}) (\boldsymbol{f}_i - \overline{\boldsymbol{f}})^{\mathrm{T}}, \tag{10}$$

where $\overline{\boldsymbol{f}}$ is the mean vector of all these training descriptors.

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

$$\mathbf{MV} = \mathbf{VD}, \tag{11}$$

where $\mathbf{D}$ is a diagonal matrix with diagonal entries equal to the eigenvalue of $\mathbf{M}$, $\mathbf{V}$ is a matrix with columns equal to the eigenvectors of $\mathbf{M}$.

We define the PCA subspace using the eigenvectors corresponding to the highest n eigenvalues. The value of n is chosen such that 95% of the fidelity is preserved in the compressed data.

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

$$\widehat{\boldsymbol{f}}_i = \mathbf{V}_n^{\mathrm{T}} \boldsymbol{f}_i, \tag{12}$$

where $\mathbf{V}_n^{\mathrm{T}}$ is the transpose of the first n columns of $\mathbf{V}$.

During feature matching, we use the Euclidean distance to measure the distance between any two descriptors $\widehat{\boldsymbol{f}}_i$ and $\widehat{\boldsymbol{f}}_j$.

3 FEATURE MATCHING PERFORMANCE

We tested the performance of TriSI descriptor on the Bologna Dataset (Tombari et al., 2010). We also compared it with several state-of-the-art descriptors including Spin Image (SpinIm) (Johnson and Hebert, 1999), Normal Histogram (NormHist) (Hetzel et al., 2001), Local Surface Patches (LSP) (Chen and Bhanu, 2007) and SHOT (Tombari et al., 2010). The experimental results are presented in this section.

3.1 Experimental Setup

The Bologna Dataset (Tombari et al., 2010) contains six models and 45 scenes. The models were taken from the Stanford 3D Scanning Repository (Curless and Levoy, 1996). The scenes were generated by randomly placing different subsets of the models in order to create clutter and pose variances. The ground-truth

Table 1: Tuned parameter settings for five feature descriptors.

	Support Radius (mr)	Dimensionality	Length
SpinIm	15	15*15	225
NormHist	15	15*15	225
LSP	15	15*15	225
SHOT	15	8*2*2*10	320
TriSI	**15**	**29*1**	**29**

transformations between each model and its instances in the scenes were provided in the dataset.

We adopt the frequently used *Recall* vs 1-*Precision* Curve (RP Curve) (Ke and Sukthankar, 2004) to measure the performance of a feature descriptor. If the distance between a scene feature descriptor and a model feature descriptor is smaller than a threshold τ, this pair of feature descriptors is considered a match. Further, if the two feature descriptors come from the same physical location, this match is considered a true positive. Otherwise, it is considered a false positive. Given the total number of positives as a priori, the recall and 1-precision are calculated, as in (Ke and Sukthankar, 2004).A RP Curve can therefore, be generated by varying the threshold τ.

We first randomly selected a set of keypoints from each model (1000 keypoints in our case), and obtained their corresponding points from the scene. We then used a feature description method (e.g., TriSI) to extract a set of feature descriptors. The scene feature descriptors were finally matched against the model feature descriptors to produce a RP Curve. The parameters for all the five feature description methods were tuned by a Tuning Dataset which contains the six models and their transformed versions (obtained by resampling to $1/2$ of their original mesh resolution and adding 0.1mr Gaussian Noise). Note that, 'mr' denotes the average mesh resolution of the six models, which is used as the unit for metric parameters. We also trained the PCA subspace using the TriSI descriptors of the six models. The tuned parameter settings for all feature descriptors are shown in Table 1.

3.2 Robustness to Noise

In order to test performance of all these descriptors in the presence of noise, we added different levels of noise with standard deviations of 0.1mr, 0.3mr, and 0.5mr to the 45 scenes. We generated one RP Curve for a feature descriptor at each noise level. The RP Curves of these five descriptors are shown in Fig. 2.

It can be observed that our TriSI descriptor achieved the best performance at all levels of noise, while SHOT achieved the second best performance. The performance of all these descriptors decreased as the level of noise increased from 0.1mr to 0.5mr. However, our TriSI was more robust to noise compared to the others. It obtained a high recall of about 0.7 together with a high precision of about 0.7 even with a high level noise (with a deviation of 0.5mr), which is shown in Fig. 2(c). Moreover, the gap between SHOT and TriSI got larger when level of noise increased. This validated the strong robustness and consistency of our TriSI descriptor in the presence of noise.

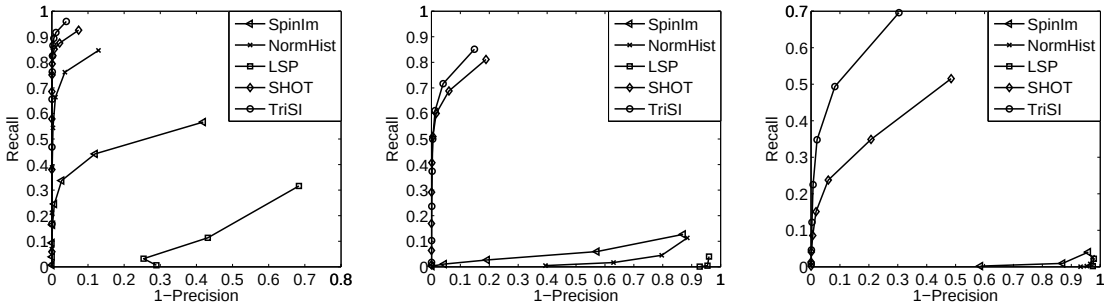
It can also be observed that NormHist and SpinIm performed well under a low level of noise (with a deviation of 0.1 mr), as shown in Fig. 2(a). They however, failed to work when a medium level of noise (with a deviation of 0.3mr) was added, as shown in Fig. 2(b). This is because the generation of a NormHist or a SpinIm requires surface normals which are very sensitive to noise. In contrast, LSP was highly susceptible to noise. It achieved a very low recall and precision even under a low level of noise (with a deviation of 0.1mr), as shown in Fig. 2(a). This is mainly due to the reason that LSP is based on the shape index values of the local surface which are even more sensitive to noise compared to surface normals.

3.3 Robustness to Varying Mesh Resolutions

In order to test the performance of these descriptors with respect to varying mesh resolutions, we resampled the noise-free scenes down to $1/2$, $1/4$ and $1/8$ of their original mesh resolution. We generated one RP Curve for a feature descriptor at each level of mesh decimation. The RP Curves of these descriptors are shown in Fig. 3.

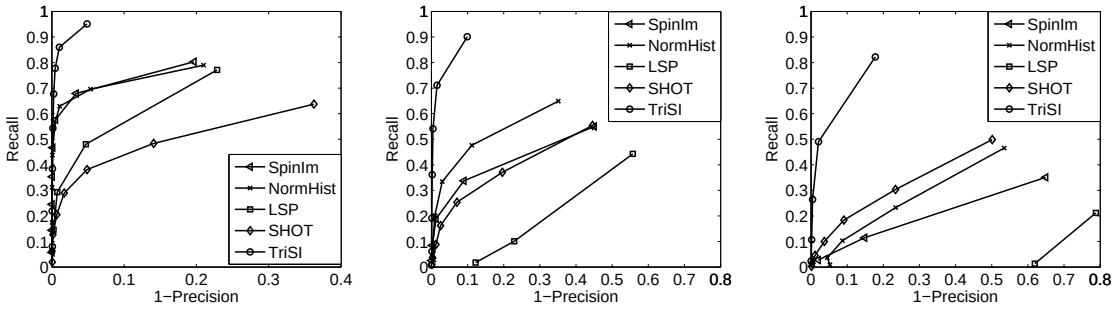
It was found that our TriSI was very robust to varying mesh resolutions, and outperformed the other descriptors by a large margin under all levels of mesh decimation. TriSI achieved a high recall of about 0.95 under $1/2$ mesh decimation, as shown in Fig. 3(a). It also achieved a recall of about 0.9 under $1/4$ mesh decimation, as shown in Fig. 3(b). TriSI consistently achieved a good performance even under $1/8$ mesh decimation, with a recall of more than 0.8, as shown in Fig. 3(c). It is worth noting that the performance of TriSI under $1/8$ mesh decimation was even better than the performance of SpinIm under $1/2$ mesh decimation, as shown in Figures 3(a) and (c). This clearly justifies the effectiveness and robustness of the TriSI descriptor with respect to varying mesh resolutions.

Moreover, our TriSI descriptor is very compact. As shown in Table 1, the length of a compressed TriSI in this experiment is only 29. In contrast, the second



(a) (b) (c)

Figure 2: Recall vs 1- precesion curves in the presence of noise. (a) Noise with a deviation of 0.1mr. (b) Noise with a deviation of 0.3mr. (c) Noise with a deviation of 0.5mr.



(a) (b) (c)

Figure 3: 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.

shortest length of the other descriptors is 225, which is larger than the length of TriSI by an order of magnitude. This means that our TriSI can be matched more efficiently compared to the other methods.

4 3D MODELING

In order to demonstrate the effectiveness of TriSI for 3D modeling, we used TriSI to perform pairwise and multiview range image registration.

4.1 Pairwise Range Image Registration

Given a pair of range images $\{\mathcal{S}_1, \mathcal{S}_2\}$ of an object, we respectively generate a set of TriSI descriptors for both $\mathcal{S}_1$ and $\mathcal{S}_2$. We match the TriSI descriptors of $\mathcal{S}_1$ against the TriSI descriptors of $\mathcal{S}_2$ to obtain a set of feature correspondences $\{\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_{N_c}\}$, where N_c is the number of feature correspondences. For each feature correspondence $\mathcal{C}_i$, we can generate a transformation $(\mathbf{R}_i, \boldsymbol{t}_i)$ between $\mathcal{S}_1$ and $\mathcal{S}_2$ using the LRFs and the point positions. That is, the rotation matrix $\mathbf{R}_i$ and translation vector $\boldsymbol{t}_i$ are calculated as follows:

$$\mathbf{R}_i = \mathbf{F}_{s1i}^{\mathrm{T}} \mathbf{F}_{s2i}, \tag{13}$$

$$\boldsymbol{t}_i = \boldsymbol{o}_{s1i} - \boldsymbol{o}_{s2i}\mathbf{R}, \tag{14}$$

where $\boldsymbol{o}_{s1i}$ and $\boldsymbol{o}_{s2i}$ are respectively the positions of the corresponding points from $\mathcal{S}_1$ and $\mathcal{S}_2$, $\mathbf{F}_{s1i}$ and $\mathbf{F}_{s2i}$ are respectively the LRFs of $\boldsymbol{o}_{s1i}$ and $\boldsymbol{o}_{s2i}$.

We calculate a total of N_c transformations from the N_c feature correspondences. These transformations are then grouped into a few clusters. Each cluster center gives a potential transformation between $\mathcal{S}_1$ and $\mathcal{S}_2$. We sort these clusters based on their sizes and their average feature distances. We verify several most likely transformations and choose the transformation which produce the least registration error between $\mathcal{S}_1$ and $\mathcal{S}_2$.

We applied our TriSI based pairwise range image registration method to two range images of the Chef model of the UWA Dataset (Mian et al., 2006). Fig. 4(a) shows the feature matching results between the two range images. It is clear that the majority of feature matches are true positives. This feature matching result further indicates the high descriptiveness of our TriSI descriptor. Although there are a few false positives in the feature matching result, they can easily be

(a) (b)

Figure 4: Pairwise range image registration results. (a) Feature matching result between two range images of the Chef. (b) Pairwise registration result. (Figure best seen in color.)

(a) (b) (c)

Figure 5: Multiview range image registration result. (a) Initial set of 22 range images of the Chef model. (b) Coarse registration result by TriSI based method. (c) The fine 3D model in the UWA Dataset. (Figure best seen in color.)

eliminated by the process of grouping. As shown in Fig 4(b), the alignment between the two range images are very accurate.

4.2 Multiview Range Image Registration

Given a set of range images $\{\mathcal{S}_1, \mathcal{S}_2, \cdots, \mathcal{S}_{N_r}\}$ of an object, we generate a set of TriSI descriptors for each range image. Once the TriSI descriptors are obtained, we use a tree based algorithm to perform multiview range image registration. We first select the range image $\mathcal{S}_i$ which has the maximum surface area as the root node of the tree. We then use the aforementioned pairwise registration method to match $\mathcal{S}_i$ with the remaining range images in the search space. Once a range image $\mathcal{S}_j$ is accurately registered with $\mathcal{S}_i$, the range image $\mathcal{S}_j$ is added to the tree as a new node and removed from the search space. The arc between the two nodes represents the rigid transformation between $\mathcal{S}_j$ and $\mathcal{S}_i$. Once all range images have been matched with $\mathcal{S}_i$, the algorithm proceeds to the next node. This process continues until all range images in the search space are added to the tree. Once the tree is fully constructed, the transformation between any two connected nodes is already available. Based on these estimated transformations, we transform all range images to the coordinate basis of the range image at the root node. These multiview range images can therefore be aligned without any manual intervention.

We applied our TriSI based multiview range image registration method to 22 range images of the Chef model of the UWA Dataset. Fig. 5(a) shows the initial set of range images, and Fig. 5(b) illustrates our registration result. It is clear that our TriSI based method achieved a highly accurate alignment without use of any priori information (e.g., the image order or initial pose of each image). A visual comparison between the coarse registered range images and the original fine 3D model shows that the registration result is almost identical to the original model, as shown in Figures 5 (b) and (c). This result further demonstrates the effectiveness of our TriSI based method. It is expected that the registration results can further be improved by using a global registration algorithm, e.g., (Williams and Bennamoun, 2001).

5 OBJECT RECOGNITION

To further evaluate the performance of our TriSI descriptor, we used TriSI to perform 3D object recognition on the UWA Dataset (Mian et al., 2006). The UWA Dataset is one of the most frequently used datasets. It contains five models and 50 real scenes.

Our 3D object recognition algorithm goes through two phases: offline pre-processing and online recognition. During the offline pre-processing phase, we extract our TriSI descriptors from a set of randomly selected keypoints from all models. We also index these TriSI descriptors using a *k*-d tree structure for efficient feature matching. During the online recognition phase, we first extract TriSI descriptors from a set of randomly selected keypoints in a given scene. We then match the scene descriptors against the model descriptors using the *k*-d tree. The feature matching results are used to vote for candidate models and to generate transformation hypotheses. The candidate models are then verified in turn by aligning them with the scene using a transformation hypothesis. If the candidate model is aligned accurately with a portion of the scene, the candidate model and transformation hypothesis are accepted. Subsequently, the scene points that correspond to this model are recognized and segmented. Otherwise, the transformation hypothesis is rejected and the next hypothesis is verified by turn.

We conducted our experiments using the same data and experimental setup as in (Mian et al., 2006) and (Bariya et al., 2012) to achieve a rigorous comparison. The recognition rates of our TriSI based method are shown in Fig. 6 with respect to varying levels of occlusion. We also present the recognition results of Tensor (Mian et al., 2006), SpinIm (Mian et al., 2006), VD-LSD (Taati and Greenspan, 2011) and EM (Bariya et al., 2012) based methods.

As shown in Fig. 6, our method achieved the best recognition results. The average recognition rate of our TriSI based methods was 98.4%. That is, only three out of the 188 objects in the 50 scenes were not correctly recognized. The second best results were produced by EM based method with an average recognition rate of 97.5%. They were followed by Tensor, VD-LSD and SpinIm based methods.

Our TriSI based method is also robust to occlusion. It achieved a 100% recognition rate under upto 80% occlusion. As occlusion further increased, the recognition rate decreased slightly. It obtained a recognition rate of more than 80% even under 90% occlusion. In contrast, the second best recognition rate under 90% occlusion was only about 50%, which was reported by the EM based method.

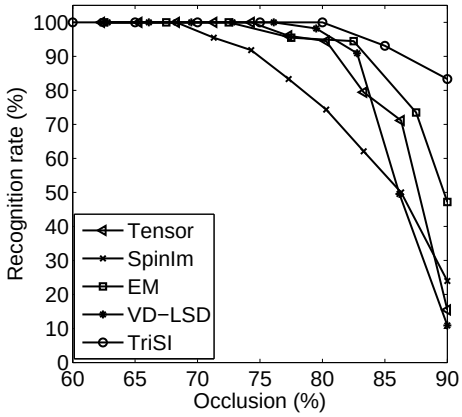


Figure 6: Recognition rates on the UWA Dataset.

6 CONCLUSIONS

In this paper, we presented a distinctive TriSI local descriptor. We evaluated the descriptiveness and robustness of our TriSI descriptor with respect to different levels of noise and mesh resolutions. Experimental results show that TriSI is very robust to noise and varying mesh resolutions. It outperformed all existing method. We also used our TriSI descriptor for both pairwise and multiview range image registration. TriSI achieved highly accurate registration results. Moreover, we used TriSI based method to perform 3D object recognition. Experimental results revealed that our method achieved the best recognition rate. Overall, our TriSI based method is robust to noise, occlusion and varying mesh resolutions. It outperformed the state-of-the-art methods.

ACKNOWLEDGEMENTS

This research is supported by a China Scholarship Council (CSC) scholarship (2011611067) and Australian Research Council grants (DE120102960, DP110102166).

REFERENCES

Attene, M., Marini, S., Spagnuolo, M., and Falcidieno, B. (2011). Part-in-whole 3D shape matching and docking. *The Visual Computer*, 27(11):991–1004.

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

Boyer, E., Bronstein, A., Bronstein, M., Bustos, B., Darom, T., Horaud, R., Hotz, I., Keller, Y., Keustermans, J.,

Kovnatsky, A., et al. (2011). SHREC 2011: Robust feature detection and description benchmark. In *Eurographics Workshop on Shape Retrieval*, pages 79–86.

Bronstein, A., Bronstein, M., Bustos, B., Castellani, U., Crisani, M., Falcidieno, B., Guibas, L., Kokkinos, I., Murino, V., Ovsjanikov, M., et al. (2010). SHREC 2010: robust feature detection and description benchmark. In *Eurographics Workshop on 3D Object Retrieval*, volume 2, page 6.

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

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

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

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

Guo, Y., Bennamoun, M., Sohel, F., Wan, J., and Lu, M. (2013). 3D free form object recognition using rotational projection statistics. In *IEEE 14th Workshop on the Applications of Computer Vision. In press*.

Guo, Y., Wan, J., Lu, M., and Niu, W. (2012). A parts-based method for articulated target recognition in laser radar data. *Optik. http://dx.doi.org/10.1016/j.ijleo.2012.08.035*.

Hetzel, G., Leibe, B., Levi, P., and Schiele, B. (2001). 3D object recognition from range images using local feature histograms. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages II–394.

Huber, D., Carmichael, O., and Hebert, M. (2000). 3D map reconstruction from range data. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 891–897. IEEE.

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

Ke, Y. and Sukthankar, R. (2004). PCA-SIFT: A more distinctive representation for local image descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 498–506.

Lai, K., Bo, L., Ren, X., and Fox, D. (2011a). A large-scale hierarchical multi-view GRB-D object dataset. In *IEEE International Conference on Robotics and Automation*, pages 1817–1824.

Lai, K., Bo, L., Ren, X., and Fox, D. (2011b). A scalable tree-based approach for joint object and pose recognition. In *Twenty-Fifth Conference on Artificial Intelligence (AAAI)*.

Mian, A., Bennamoun, M., and Owens, R. (2005). Automatic correspondence for 3D modeling: An extensive review. *International Journal of Shape Modeling*, 11(2):253.

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

Mian, A., Bennamoun, M., and Owens, R. (2010). 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.

Novatnack, J. and Nishino, K. (2008). Scale-dependent/invariant local 3D shape descriptors for fully automatic registration of multiple sets of range images. In *10th European Conference on Computer Vision*, pages 440–453.

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

Salti, S., Tombari, F., and Stefano, L. (2011). A performance evaluation of 3D keypoint detectors. In *International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, pages 236–243.

Shilane, P., Min, P., Kazhdan, M., and Funkhouser, T. (2004). The Princeton shape benchmark. In *International Conference on Shape Modeling Applications*, pages 167–178.

Sun, Y. and Abidi, M. (2001). Surface matching by 3D point's fingerprint. In *8th IEEE International Conference on Computer Vision*, volume 2, pages 263–269.

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

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

Williams, J. and Bennamoun, M. (2000). A multiple view 3D registration algorithm with statistical error modeling. *IEICE Transactions on Information and Systems*, 83(8):1662–1670.

Williams, J. and Bennamoun, M. (2001). Simultaneous registration of multiple corresponding point sets. *Computer Vision and Image Understanding*, 81(1):117–142.