



# Keypoints-based surface representation for 3D modeling and 3D object recognition



Syed Afaq Ali Shah<sup>a,\*</sup>, Mohammed Bennamoun<sup>a</sup>, Farid Boussaid<sup>b</sup>

<sup>a</sup> School of Computer Science and Software Engineering, The University of Western Australia, Perth 6009, Australia

<sup>b</sup> School of Electrical, Electronics and Computer Engineering, The University of Western Australia, Perth 6009, Australia

## ARTICLE INFO

### Keywords:

3D modeling

3D Object recognition

Range images

Keypoints

## ABSTRACT

The three-dimensional (3D) modeling and recognition of 3D objects have been traditionally performed using local features to represent the underlying 3D surface. Extraction of features requires cropping of several local surface patches around detected keypoints. Although an important step, the extraction and representation of such local patches adds to the computational complexity of the algorithms. This paper proposes a novel Keypoints-based Surface Representation (KSR) technique. The proposed technique has the following two characteristics: (1) It does not rely on the computation of features on a small surface patch cropped around a detected keypoint. Rather, it exploits the geometrical relationship between the detected 3D keypoints for local surface representation. (2) KSR is computationally efficient, requiring only seconds to process 3D models with over 50,000 points with a MATLAB implementation. Experimental results on the UWA and Stanford 3D models dataset suggest that it can accurately perform pairwise and multiview range image registration (3D modeling). KSR was also tested for 3D object recognition with occluded scenes. Recognition results on the UWA dataset show that the proposed technique outperforms existing methods including 3D-Tensor, VD-LSD, keypoint-depth based feature, spherical harmonics and spin image with a recognition rate of 95.9%. The proposed approach also achieves a recognition rate of 93.5% on the challenging Ca'Fascori dataset compared to 92.5% achieved by game-theoretic. The proposed method is computationally efficient compared to state-of-the-art local feature methods.

## 1. Introduction

The computation of similarity between three dimensional (3D) surfaces is key to a number of pattern recognition tasks such as 3D modeling and 3D object recognition [1–3]. The aim of 3D modeling is to measure the similarity between 3D surfaces captured from different viewpoints, align and merge them to construct a complete 3D model of an object [4–7]. On the other hand, the task of 3D object recognition consists in correctly determining the identity and pose of objects in a scene [8–10]. Both these tasks find vast applications in fields such as robotics [11–13], reverse engineering [14], scene understanding [15,16], medical [17,18] and biometric systems [19,20].

For the last two decades, the most popular approach to measure the similarity between surfaces (for 3D modeling and object recognition) exploits a compact representation of the 3D surface, known as 3D features [21,22]. Local correspondences established by matching the features are used to solve higher level tasks such as automatic 3D modeling and 3D object recognition. Local surface description can address the challenges posed by changes in viewpoint, clutter and

occlusion [8,23]. A variety of 3D surface features have been proposed in the literature. Mian et al. [8] proposed a 3D-Tensor descriptor for 3D object recognition. A local reference frame was first constructed by selecting a pair of vertices that satisfied certain geometric constraints. A 3D-tensor descriptor was then generated by constructing a local 3D grid over the range image, and summing the surface areas intersecting each bin of the 3D grid. Osada et al. [24] proposed shape distribution. In their proposed technique, the shape signature for a 3D model is represented as a probability distribution sampled from a shape function measuring surface properties of the 3D model. Their global representation relies on five shape functions, which include: angle between three random points, a distance between a fixed point and a random point, distance between two random points, the square root of the area of the triangle between three random points and the cube root of the volume of the tetrahedron between four random points on the 3D surface. These five shape functions are then used to define their proposed shape distribution, which follows a complex process of iteration through all the vertices of a 3D mesh. In addition, careful selection of few parameters such as N samples, B Number of bins and V

\* Corresponding author.

E-mail address: [syed.shah@research.uwa.edu.au](mailto:syed.shah@research.uwa.edu.au) (S.A.A. Shah).

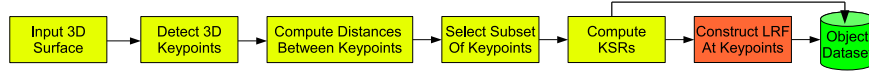


Fig. 1. Proposed methodology for computing Keypoints based Surface Representation (KSR).

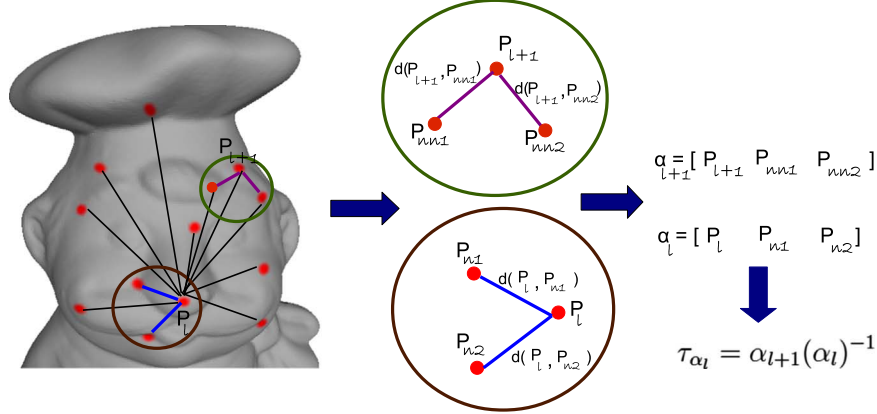


Fig. 2. Illustration of the proposed Keypoint based Surface Representation (KSR) technique. The colored lines show the distance measured between the keypoint  $P_l$  and the other detected keypoints. The two nearest neighboring keypoints of the  $P_l$  are selected based on their minimum distance (shown in blue) to  $P_l$ . The same procedure is repeated for keypoint  $P_{l+1}$ . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

number of vertices is also required for a good representation. Johnson and Hebert [25] used the normal  $n$  of a keypoint  $p$  as the local reference axis to generate a “spin image” descriptor. They expressed each neighboring point  $q$  with two parameters: the radial distance  $\rho$  and the signed distance  $\varphi$ . They then discretized the  $\rho$ - $\varphi$  space into a 2D array accumulator, and counted up the number of points that fell into the bin indexed by  $(\rho, \varphi)$ . The 2D array was further bilinearly interpolated to construct the spin image. Taati et al. [26] proposed a Variable Dimensional Local Shape Descriptor (VD-LSD). They first performed PCA on the covariance matrix of the neighboring points of each point  $q$  on the surface. Therefore, each point  $q$  had a local reference frame and three eigenvalues  $(\lambda_1, \lambda_2, \lambda_3)$ . They then calculated a set of position properties, direction properties and dispersion properties for a point  $q$ . The position properties of  $q$  included its 3D coordinates expressed in a local reference frame. The direction properties included the Euler angles which were used to register the local reference frame with the global frame. The dispersion properties included the three eigenvalues  $(\lambda_1, \lambda_2, \lambda_3)$ . They then selected a subset of these properties using a feature selection algorithm. Finally, they accumulated these selected properties of all the neighboring points of a keypoint  $p$  into a histogram (i.e., VD-LSD). Tombari et al. [27] proposed a descriptor named Signature of Histograms of Orientations (SHOT). They first constructed a local reference frame for a keypoint  $p$ , and divided the neighborhood space into 3D spherical volumes. They then generated a local histogram for each volume by accumulating the number of points according to the angles between the normal at the keypoint and the normals at the neighboring points. They concatenated all local histograms to form an overall SHOT descriptor. The SHOT descriptor is highly descriptive, computationally efficient and robust to noise. Experimental results showed that SHOT outperformed the spin image and point signature at all levels of noise. Darom et al. proposed Scale Invariant Spin Image (SISI) and Local Depth SIFT (LD-SIFT) descriptors for 3D mesh models [28]. In their proposed technique, the SISI descriptor is constructed by computing the spin image descriptor over local scale, while LD-SIFT is computed by representing the vicinity of the keypoints as a depth map.

Although existing local feature based techniques are accurate and can handle occlusion and clutter, these approaches still suffer from high computational complexity [23,29]. To overcome these shortcomings, we propose a novel technique, called Keypoint based Surface Representation (KSR), which does not require computation of local features, around each detected keypoint using a local surface patch.

Most of the existing feature extraction techniques crop a local surface around a detected keypoint prior to defining a local feature [21,23]. In contrast, the proposed KSR exploits geometrical relationships between detected keypoints for reliable local surface representation. The proposed KSR can be used with any state-of-the-art keypoint detector which is the major advantage of our approach. The proposed technique is also computational efficient and achieves state-of-the-art 3D modeling and object recognition performance.

The rest of this paper is organized as follows. Section 2 describes the proposed Keypoints based Surface Representation (KSR) technique. The performance evaluation of the proposed technique with state-of-the-art keypoint detectors, its robustness to noise and variations in mesh resolution are presented in Section 3. Section 4 presents our pairwise registration and 3D modeling algorithms based on KSR. The qualitative and quantitative evaluation of the 3D modeling accuracy is presented in Section 5. Section 6 presents our automatic 3D object recognition algorithm. Section 7 reports our object recognition results and comparisons with state-of-the-art recognition techniques. This is followed by a computational efficiency analysis of the proposed technique. The paper is concluded in Section 8.

## 2. Keypoints based surface representation (KSR)

The proposed methodology for computing the Keypoints based Surface Representation (KSR) is depicted in Fig. 1, while Fig. 2 illustrates the proposed technique. Briefly stated, the KSR method operates as follows. In the first step, 3D keypoints are detected. Next, the geometrical relationships between keypoints are computed by measuring the distances between them. Then, subsets of keypoints are selected based on the minimum distance between them, as illustrated in Fig. 2. KSR is finally computed using these subsets of keypoints (Fig. 2). A Local Reference Frame (LRF) is then defined at each keypoint. Finally, all the keypoints along with their KSRs and LRFs are stored in the object database for later use in the 3D modeling and object recognition tasks. The details of each step are given below.

For a given 3D surface  $\gamma_1$ , the keypoints  $P_k, k = 1, 2, \dots, K$  ( $K$  being the total number of keypoints) are detected using the recently proposed DoG based keypoint detector [28]. DoG algorithm detects highly repeatable keypoints and keeps the same order of the keypoints for a given 3D surface. Note that any other state-of-the-art keypoint detector including Harris 3D [30], keypoint quality [31,32] can also be used in this case, as demonstrated in Section 3. For a given 3D keypoint (e.g.,

$P_l$ ), its geometrical distance to all the keypoints on the 3D surface (Fig. 2) is measured as:

$$d(P_l, P_k) = \sqrt{\sum_{k=2}^K (P_l - P_k)^2} \quad (1)$$

Based on the minimum geometrical distance between the keypoints, the two nearest neighboring keypoints of  $P_l$  are selected to form a subset of keypoints  $\alpha_l$ , with  $\alpha_l \subset P_k$ . The two nearest neighboring points are selected empirically to keep a good tradeoff between accuracy and computational efficiency.  $\alpha_l$  can be represented as follows:

$$\alpha_l = [P_{n_1} P_{n_2}] \quad (2)$$

where  $P_{n_1}$  and  $P_{n_2}$  are the two nearest neighboring keypoints of  $P_l$ . Eq. (2) can also be written as:

$$\alpha_l = \begin{bmatrix} x_{P_l} & x_{P_{n_1}} & x_{P_{n_2}} \\ y_{P_l} & y_{P_{n_1}} & y_{P_{n_2}} \\ z_{P_l} & z_{P_{n_1}} & z_{P_{n_2}} \end{bmatrix} \quad (3)$$

It can be noted that each column of  $\alpha_l$  contains the 3D coordinates  $x, y, z$  of the three keypoints, respectively. The keypoints subsets  $\alpha_{l+1}, \dots, \alpha_k$  for keypoints  $P_{l+1}, \dots, P_k$  are computed in a similar manner.

The transformation between the subsets  $\alpha_l$  and  $\alpha_{l+1}$  is then computed as follows:

$$\alpha_{l+1} = \tau_{\alpha_l} \alpha_l \quad (4)$$

$$\tau_{\alpha_l} = \alpha_{l+1} (\alpha_l)^{-1} \quad (5)$$

where  $\alpha_{l+1} = [P_{l+1} P_{nn_1} P_{nn_2}]$ ,  $P_{nn_1}$  and  $P_{nn_2}$  are the two nearest neighboring keypoints of  $P_{l+1}$  (Eq. (1)).  $\tau_{\alpha_l}$  computes the geometrical relationship between subset of keypoints  $\alpha_l$  and  $\alpha_{l+1}$  to form the proposed KSR. Algorithm 1 details our KSR algorithm. In a similar way, relationships are computed between the remaining subsets of keypoints  $(\alpha_{l+1}, \alpha_{l+2}, \dots, \alpha_k)$  for  $\gamma_1$ , to get the final set of KSRs  $\tau_\alpha = \tau_{\alpha_1}, \tau_{\alpha_2}, \dots, \tau_{\alpha_{K-1}}$ . Similarly, for 3D surface  $\gamma_2$ , we get  $\tau_\beta = \tau_{\beta_1}, \tau_{\beta_2}, \dots, \tau_{\beta_L}$  ( $L = J - 1$ ), where  $J$  is the total number of keypoints on  $\gamma_2$ .

To handle the scenario of singularity of  $\alpha_l$  or  $\beta_L$  that would impeditment the computation of matrix inverse, we use popular Moore-Penrose pseudoinverse [33,34] to calculate the inverse of a matrix.

**Algorithm 1.** Keypoint based Surface Representation Algorithm.

**Input:** Input 3D image:  $\gamma_1$ ,  
 $P_k \leftarrow$  detected keypoints  
1 For a given keypoint  $P_l$   
2 // Measure distance between  $P_l$  and other keypoints  $P_k$   
3 **for** each keypoint  $P_k, k=2, \dots, K$  **do**  
4  $d(P_l, P_k) \leftarrow \sqrt{\sum_{k=2}^K (P_l - P_k)^2}$   
5 **end**  
6 // Based on minimum geometrical distances, select subset of keypoints  
7  $\alpha_l \leftarrow [P_l P_{n_1} P_{n_2}]$   
8 // Select subset of keypoints for  $P_{l+1}$  using above procedure  
9  $\alpha_{l+1} \leftarrow [P_{l+1} P_{nn_1} P_{nn_2}]$   
10 // Compute KSR  
11  $\tau_{\alpha_l} \leftarrow \alpha_{l+1} (\alpha_l)^{-1}$   
**output** KSR:  $\tau_{\alpha_1}, \tau_{\alpha_2}, \dots, \tau_{\alpha_{K-1}}$  for all the subsets of keypoints.

**KSRs Ambiguity:** Once all the KSRs have been computed, there may be a possibility that two or more KSRs on a given range image are similar. To eliminate these ambiguities, we measure the similarity

between KSRs (e.g.,  $\tau_{\alpha_i}$ ) as follows:

$$err = \cos^{-1} \left( \tau_{\alpha_i} \left( \tau_{\alpha_j} \right)^T \right) \quad (6)$$

where the term between the brackets is the dot product of the two vectors and  $err$  represents their mutual angle. The value of  $err$  may vary from zero (perfectly similar vectors) to  $\pi/2$  (completely dissimilar). The KSR vectors which give the minimum error  $err$  are considered a match and the duplicate KSR is eliminated from the list of KSRs  $\tau_{\alpha_i}$ . All the remaining KSRs are then stored in the object database. Experimental results on the UWA dataset show that the similarity ambiguity was found only in less than 1% of the total cases.

Once all the KSRs have been computed, we use correlation coefficient to match a pair of KSRs. The main reason of using correlation coefficient is that being a statistical measure, correlation coefficient performs better in the presence of noise compared to other matching techniques (e.g., linear matching). Details about the matching process are provided in Section 6.3.

We next construct a unique rotation invariant Local Reference Frame (LRF) at each keypoint using the technique proposed in [23]. The computed LRF is also stored in the object database for later use in the tasks of 3D modeling and object recognition. Note that the computation of KSR and LRF construction can be done in parallel to achieve a higher computational efficiency.

### 3. Performance evaluation of KSR

The performance of the proposed technique with different keypoint detectors was evaluated on the Bologna dataset [27], for the task of object recognition. The robustness of KSR was also tested with respect to different levels of noise and varying mesh resolutions. In the following, we briefly describe the dataset, keypoint detectors and the evaluation criteria used for the proposed technique.

#### 3.1. Bologna dataset

The Bologna Dataset comprises 45 scenes and six models which include, Armadillo, Asia Dragon, Bunny, Dragon, Happy Buddha and Thai Statue taken from the Stanford 3D Scanning Repository. Each scene is synthetically generated by randomly rotating and translating three to five models in order to create clutter and pose variations. As a result, the ground truth rotations and translations between each model and its instances in the scenes were known a priori during the construction process.

#### 3.2. Keypoint detectors

To demonstrate that the proposed technique can be used with any existing keypoint detector, we evaluated KSR with several popular state-of-the-art detectors. These include DoG detector [28], Keypoint [31], MeshDoG [35], Local Surface Patches (LSP) [36], Intrinsic Shape Signatures (ISS) [37] and Heat Kernel Signature (HKS) [38]. In addition to these detectors, we also used randomly selected points as keypoints to evaluate KSR.

#### 3.3. Evaluation criteria

The performance of the proposed technique was tested using the popular criterion of recall versus 1-precision curve (RP curve) that has been extensively used for the evaluation of local features [23,39,40].

Ideally, a RP Curve would fall in the top left corner of the plot, which means that the feature obtains both high recall and precision.

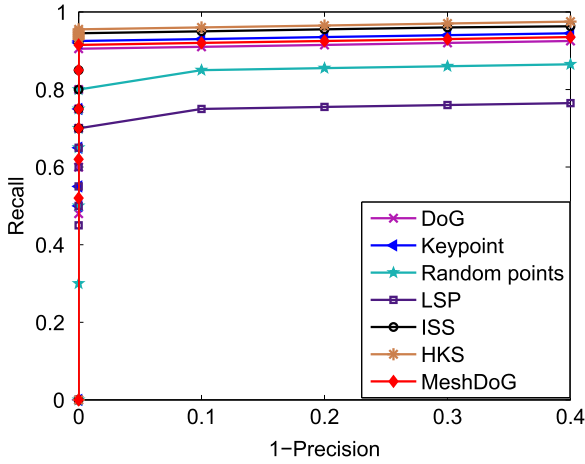


Fig. 3. Recall vs 1-precision curves. Performance of KSR with state-of-the-art keypoint detectors.

### 3.4. KSR Performance with different Keypoint Detectors

To evaluate the performance of KSR with different state-of-the-art keypoint detectors (Section 3.2), we detected 3D keypoints using these techniques. KSR was then computed using the detected keypoints. The RP curves for KSR are presented in Fig. 3. The proposed technique achieves comparable performance with all the keypoint detectors, as shown in Fig. 3. These results suggest that the proposed technique achieves good performance with existing keypoint detectors by achieving a high recall and precision for all these techniques.

### 3.5. KSR performance under noise variations

In order to evaluate the performance of KSR for different noise levels, we added a Gaussian noise with standard deviations of 0.1, 0.3 and 0.5mr to the scene data. The RP curves under different levels of noise are presented in Fig. 4(a)–(c).

Note that with noise of 0.1mr, the proposed technique achieves the highest performance with HKS followed by MeshDoG and DoG (Fig. 4(c)). As the noise level increases to 0.3mr, the proposed technique still achieves the highest performance with HKS, while the performance with keypoint, DoG, MeshDoG and ISS is comparable (Fig. 4(b)). For the same noise level, the performance deteriorates with LSP and random points. For a noise level of 0.5mr, the proposed technique achieves the best performance with DoG detector followed by ISS, as shown in Fig. 4(c). These results are expected for the DoG detector given that the Difference of Gaussian (DoG) algorithm removes high frequency components (e.g. random noise). As a result, it is the most suitable for processing 3D information with a high level of noise.

### 3.6. KSR Performance under varying mesh resolutions

To evaluate the performance of the proposed technique with these keypoint detectors for varying mesh resolutions, we resampled the scene meshes to  $\frac{1}{2}$ ,  $\frac{1}{4}$  and  $\frac{1}{8}$  of their original mesh resolution. The RP Curves under different levels of mesh decimation are presented in Fig. 5.

For a mesh decimation of 1/2, the proposed KSR achieves the best performance with DoG, followed by MeshDoG and Keypoint/depth based approach. As the downsampling is increased to 1/4, the performance of KSR with these keypoint detectors decreases. Under 1/4 mesh decimation, KSR achieves the best performance with the DoG detector. The second best is achieved with MeshDoG. Finally, for a mesh decimation of 1/8, KSR still achieves the best performance with DoG, followed by Keypoint, HKS and ISS.

In summary, these results suggest that the proposed KSR can be used with any existing keypoint detector. The proposed technique is robust to variations in noise and mesh resolutions.

## 4. 3D modeling

To evaluate the performance of the proposed technique, we used KSR to perform range image registration under two different scenarios: a pairwise registration (where only two 3D views are used) and a multiview registration, in which, multiple views of the object acquired from different viewpoints and are made available in no particular order.

### 4.1. Pairwise range image registration

Given a pair of range images  $\{\gamma_1, \gamma_2\}$  of an object and their set of KSRs  $\tau_\alpha$  and  $\tau_\beta$  respectively.  $\tau_\alpha$  is matched with  $\tau_\beta$  to obtain a set of correspondences  $C_1, C_2, \dots, C_{N_c}$ , where  $N_c$  is the number of correspondences. We used the correlation coefficient  $C_c$  for matching a pair of KSRs. For each correspondence  $C_i$ , we calculate the rotation matrix  $R$  and translation vector  $t$  as follows:

$$R = LRF_1^T LRF_2 \quad (7)$$

$$t = p_1 - p_2 R \quad (8)$$

$$\gamma_{12} = \gamma_1 \cup (\gamma_2 R + t) \quad (9)$$

where  $p_1$  and  $p_2$  are the positions of corresponding points from  $\gamma_1$  and  $\gamma_2$  respectively.  $LRF_1$  and  $LRF_2$  are respectively the LRFs of  $p_1$  and  $p_2$ . In Eq. (9),  $\gamma_{12}$  is the union of two meshes after the vertices of  $\gamma_2$  are rotated and translated using  $R$  and  $t$  from Eqs. (7) and (8) respectively. Fig. 6(a) and (b) illustrate the feature matching process between two different views and their pairwise registration, respectively, achieved by aligning two range images of the chef model. As shown in Fig. 6(b), the alignment between the two range images is very accurate. Note that no refinement using e.g., ICP algorithm [41] has been performed in this case.

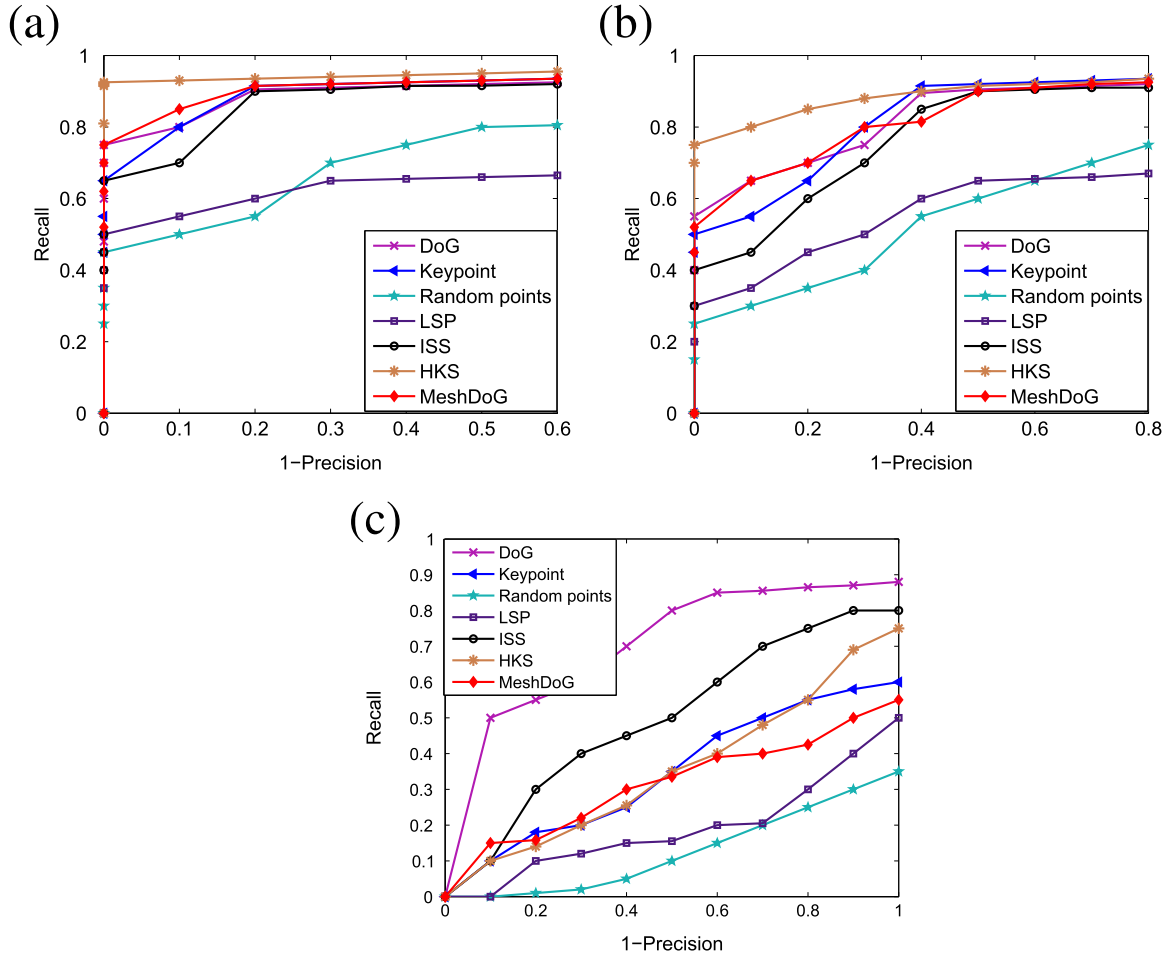
### 4.2. Multiview range image registration

To demonstrate the effectiveness of the proposed technique for 3D modeling, we next performed automatic multiview registration of range images  $\{\gamma_1, \gamma_2, \dots, \gamma_N\}$  of a 3D object. We used a tree based approach to perform multiview range image registration. We selected range image  $\gamma_i$  as the root node of the tree. We then used the aforementioned pairwise range image registration technique (Section 4.1) to match the KSRs  $\tau_\alpha$  and to register  $\gamma_i$  with  $\gamma_j$ . Once the range image  $\gamma_j$  is accurately registered with  $\gamma_i$ , the range image  $\gamma_j$  is added to the tree as a new node. The rigid transformation between  $\gamma_i$  and  $\gamma_j$  is represented by an arc between the two nodes. Once the tree is fully constructed by registering all the range images, the transformation between any two nodes is then made available. Based on these estimated transformations, all range images are transformed to the same coordinate system of the range image at the root node. These multiview range images can therefore be aligned without any user/manual intervention.

## 5. 3D modeling accuracy

### 5.1. Experimental setting

To evaluate the accuracy of 3D modeling, we performed a qualitative and a quantitative evaluation of the proposed technique on the UWA [8] and Stanford 3D models dataset [42]. The UWA dataset is a popular and widely used dataset for 3D modeling. It contains 16–22 2.5D views of four different objects. Similarly, the Stanford 3D models dataset is also one of the renowned datasets and has served as a



**Fig. 4.** Performance of KSR under noise level of (a) 0.1mr, (b) 0.3mr, (c) 0.5mr (Figure best viewed in color) (a) Noise with a standard deviation of 0.1mr, (b) Noise with a standard deviation of 0.3mr, (c) Noise with a standard deviation of 0.5mr. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

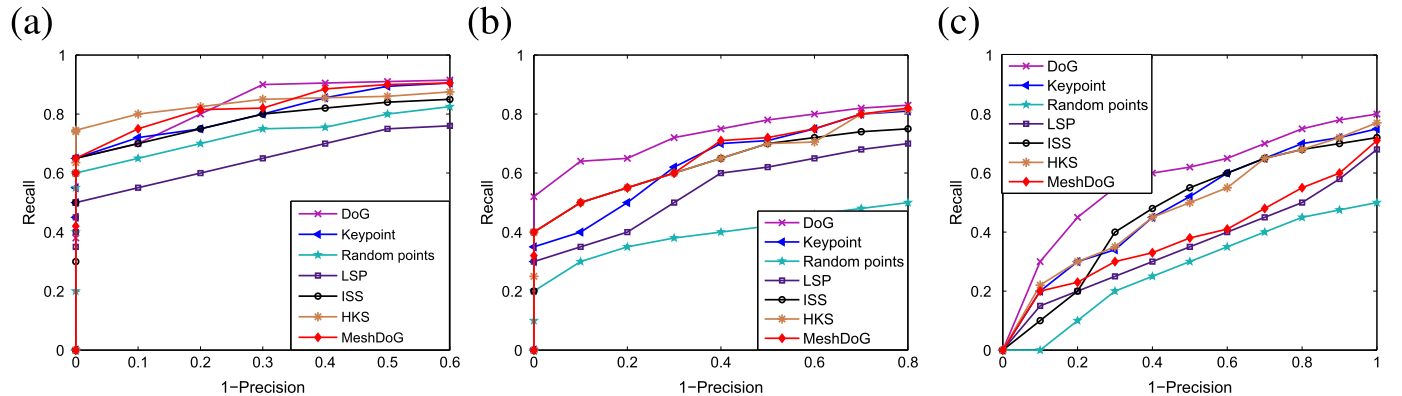
benchmark in the evaluation of 3D-Modeling algorithms [8]. It contains various 3D models of the objects (bunny being the most popular) and provides the ground truth transformations for accuracy measurement.

Performance evaluation of the proposed technique against popular spin images and SHOT descriptor is also reported in this section.

## 5.2. Qualitative evaluation

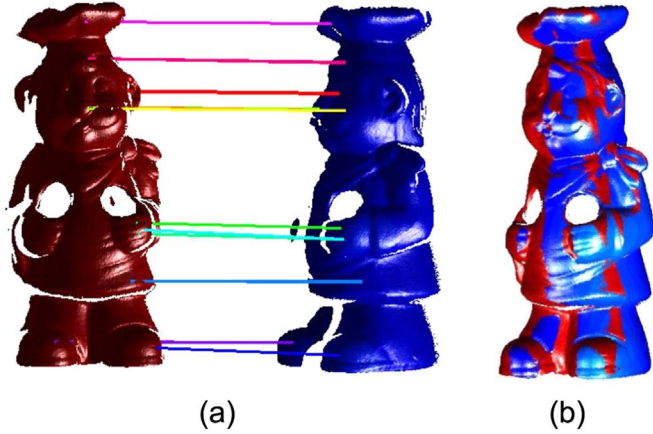
To qualitatively evaluate the proposed technique, we applied our

KSR based 3D modeling approach to range images of the UWA and Stanford 3D models datasets. Fig. 7(a) shows the initial set of range images of the chef model taken from UWA dataset. Fig. 7(b) and (c) shows the results of coarse registration. It is clear that the proposed technique achieves a highly accurate alignment of 3D views. A visual comparison between the coarsely registered range image shown in Fig. 7(b)-(c) and the original fine model displayed in Fig. 7(d) shows that the registration result is almost identical to the original model. This result further demonstrates the effectiveness of our proposed technique. It is expected that the registration results can further be



**Fig. 5.** Performance of KSR in terms of RP curves for varying mesh resolutions with existing keypoint detectors. (Figure best viewed in color) (a) 1/2 mesh decimation, (b) 1/4 mesh decimation, (c) 1/8 mesh decimation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)





**Fig. 6.** Feature matching and Registration. (a) Feature matching between two views. Only few feature correspondences are shown. (b) Registered views. This Figure is best viewed in color. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

improved by using ICP [43] and a global registration algorithm.

Fig. 8(a) shows the initial set of range images for the bunny model taken from the Stanford dataset [42], while Fig. 8(b) and (c) show the coarse registration results for the front and back view of the bunny, respectively. Fig. 8(d) shows the bunny model in a single color for visual clarity. These qualitative results suggest that the proposed technique also achieves an accurate registration on the Stanford 3D models dataset [42]. These results further demonstrate the effectiveness of our proposed technique.

### 5.3. Quantitative evaluation

To further evaluate the accuracy of our KSR based 3D modeling, we performed a quantitative evaluation of the proposed technique.

#### 5.3.1. Stanford 3D models dataset

In our first experiment, we used Stanford 3D models with available ground truth data [42]. The available ground truth rotation matrix  $R_{GT}$  and translation vector  $t_{GT}$  were recorded. Next the rotation matrix  $R_m$  and translation vector  $t_m$  resulting from our proposed technique were compared to the ground truth data. The error of the two rotation matrices was calculated using:

$$R_d = R_m R_{GT}^{-1} \quad (10)$$

$$\theta_e = \cos^{-1} \left( \frac{\text{trace}(R_d) - 1}{2} \right) \frac{180}{\pi} \quad (11)$$

In Eq. (10),  $R_d$  is the rotation matrix denoting the difference between  $R$  and  $R_{GT}$ .  $R_d$  is equal to the identity matrix in the case of zero error.  $\theta_e$  in Eq. (11) represents the amount of rotation error present in  $R$ , and is derived from Rodrigue's formula. Similarly, the translation error  $t_d$  of each view is calculated as follows:

$$t_d = \frac{\|t_m - t_{GT}\|}{mr} \quad (12)$$

In Eq. (12),  $mr$  is the mesh resolution of the 3D point-cloud. Table 1 reports the 3D-modeling accuracy achieved by the proposed technique on Stanford 3D models dataset. The proposed technique achieves a very high 3D modeling accuracy for the Stanford models. KSR achieves 100% accuracy for the models of Bunny, Armadillo and Asian Dragon. These results demonstrate the superior performance of the proposed KSR. The performance of the proposed technique is also evident from Fig. 8.

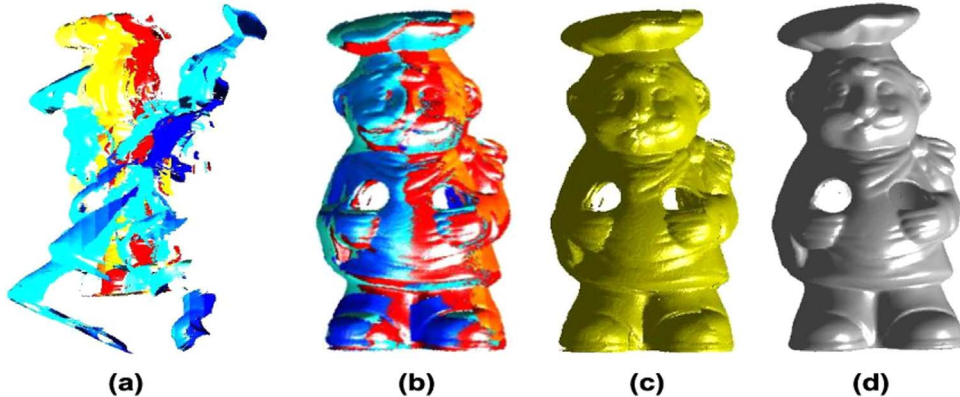
#### 5.3.2. UWA dataset

In the next experiment, each 3D model was generated using the proposed KSR technique. To measure accuracy, the four 3D models provided with the UWA dataset were used as the ground-truth models. The generated models were accurately aligned with their corresponding ground-truth models using the ICP algorithm. The residual error  $\epsilon_r$  of the ICP registration was used as the quality measure for the 3D modeling process. The modeling errors of the four objects are shown in Table 2, along with the comparison with state-of-the-art methods.

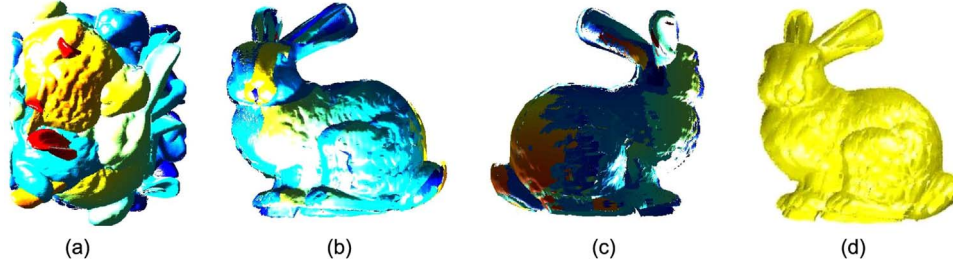
As can be noted, the difference in the modeling errors of these four objects is relatively small. The largest modeling error was  $0.7954mr$ , while the smallest modeling error was  $0.3916mr$ . Since the average modeling error ( $0.5542mr$ ) is smaller compared with the average mesh resolution, the modeling process can be considered as very accurate. The reported results suggest that the proposed technique achieves state-of-the-art performance on the UWA dataset.

## 6. 3D object recognition

In this section, we describe our fully automatic 3D object recognition algorithm, which is based on our novel KSR technique. Fig. 9 shows the block diagram of our proposed 3D object recognition. Our algorithm goes through two phases, offline training and online recognition. During the offline phase, the keypoints are first detected and KSR between keypoints are then computed for all 3D models and stored in the object dataset. During the online phase, the KSRs are computed for the given scene. The scene KSRs are then matched with the model KSRs using the linear correlation coefficient. The KSR matching results are used to vote for candidate models and generate hypotheses to transform the model to the scene. The candidate models



**Fig. 7.** Example to illustrate the results of our multiview range image registration. (a) Initial set of 11 range images of the Chef model. (b) Coarse registration result by KSR based method. The different views are shown in different colors. (c) The coarse registration result shown in one color for visual clarity. (d) The final 3D model in the UWA Dataset (Ground truth). Figure best viewed in color. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



**Fig. 8.** An example to illustrate the results of our 3D modeling approach on a Stanford 3D Model [42]. (a) An initial set of 9 range images of the Bunny model. (b) Front view of bunny. Coarse registration result by KSR based method. Different views are shown in different colors. (c) The coarse registration result for the back view of the bunny. (d) Coarse registration result shown in one color for visual clarity. Figure is best viewed in color. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

**Table 1**

3D modeling accuracy achieved by the proposed KSR on Stanford dataset. A comparison with RoPs descriptor is also provided.

Models	Bunny	Happy Buddha	Dragon	Armadillo	Lucy	Asian Dragon	Thai Statue
3D modeling accuracy (KSR)	100%	99.1%	98.5%	100%	97.3%	100%	98.2%
3D modeling accuracy (RoPs)	99.88%	100%	98.46%	97.2%	96.8%	100%	97.8%

**Table 2**

Modeling errors (in *mr*) of the four objects of the UWA dataset.

Techniques	Chef	Chicken	Parasaurolophus	T-Rex	Average
SHOT [44]	0.6696	0.4707	0.4290	1.1744	0.6859
Spin images [25]	0.9610	0.3795	0.5741	0.7688	0.6709
<b>Proposed KSR</b>	0.6205	0.3916	0.4094	0.7954	0.5542

are then verified in turn by aligning them with the scene using the hypothesis. If the candidate model is aligned accurately with a portion of the scene, the candidate and hypothesis are accepted. As a result, the scene points corresponding to this model are recognized and segmented. Otherwise, the hypothesis is rejected and the next one is verified in turn.

### 6.1. 3D models library

During the offline training phase, a model library  $M = \{M_1, M_2, \dots, M_N\}$  for the 3D objects is first constructed. For a model  $M_i$ , 3D keypoints are first detected. For each 3D keypoint, its two nearest neighboring keypoints are selected based on the minimum distance to the considered keypoint. Subsets of these keypoints (as explained in Section 2) are used to compute KSR. At each keypoint, LRF is also computed. The 3D models, the KSRs for each model and the

corresponding LRFs are then stored in the 3D models library for object recognition.

### 6.2. Scene representation

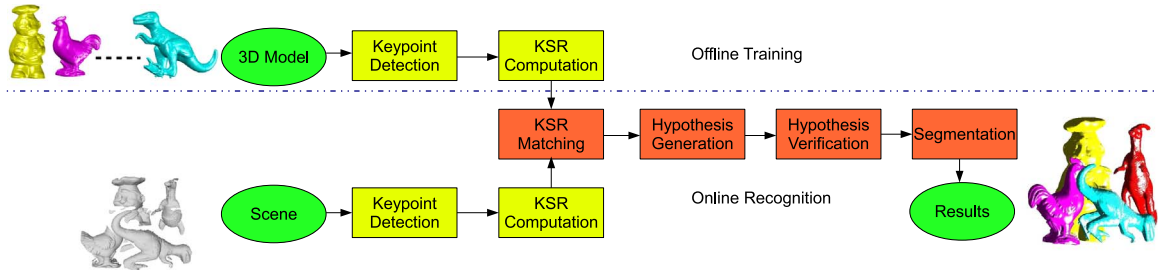
The input to our automatic 3D object recognition algorithm is a range image of a scene and the output is the recognition and segmentation of the library models in the scene. During the online recognition phase, 3D keypoints are detected on the scene range images. Next, KSRs are computed for each scene. The algorithm then moves to the feature matching phase as described below.

### 6.3. KSR matching and hypothesis generation

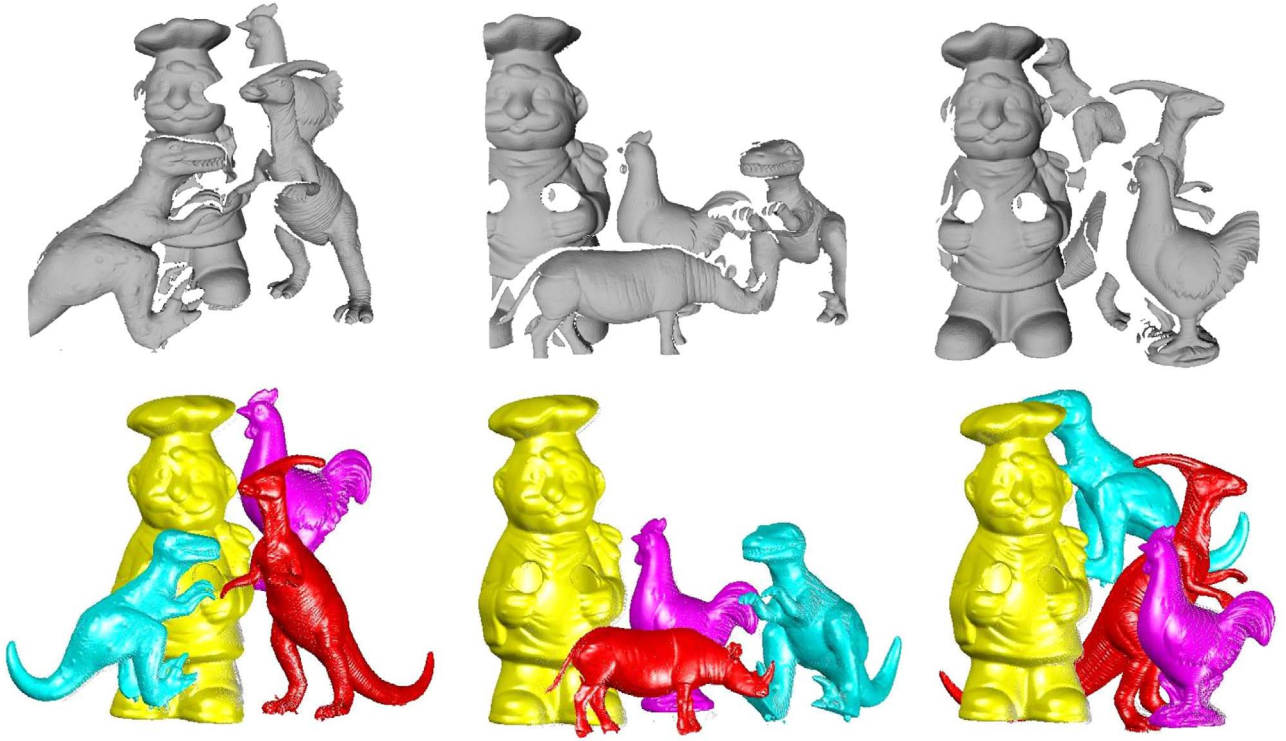
To establish correspondence between scene  $S$  and the model  $M_i$ , the overlap ratio  $R_o$  of the KSRs of  $S$  and  $M_i$  is first calculated using Eq. (13).

$$R_o = \frac{\varsigma_t}{\varsigma_s + \varsigma_m - \varsigma_t} \quad (13)$$

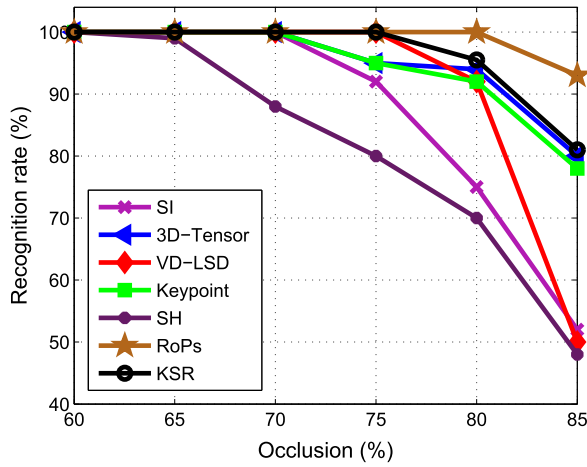
where  $\varsigma_t$  is the number of non-zero elements of KSRs from  $S$  that have a corresponding non-zero element at the same index position in the KSRs of  $M_i$ .  $\varsigma_s$  and  $\varsigma_m$  are the total numbers of non-zero elements in the KSRs of  $S$  and  $M_i$ , respectively. If  $R_o$  is greater than a pre-specified threshold  $t_R$  (equal to 0.6 in our experiments), the algorithm proceeds to calculate the correlation coefficient  $C_c$  of the two KSRs in their region of overlap. The scene and the model KSRs are considered to be the corresponding features. Otherwise, the next KSR from  $S$  is considered for matching. Each feature correspondence then votes for a model. The models, which receive votes from feature correspondences, are considered candidate models. These models are then ranked based on the number of votes they receive. The candidate models are considered model hypothesis  $\mathcal{H} = \{h_1, h_2, \dots, h_n\}$ . Each  $h_i$  is defined by a pair  $(M_h, \mathcal{T}_h)$ , where  $M_h$  is the model hypothesis and  $\mathcal{T}_h$  is the pose hypothesis, which is used to transform model  $M$  to scene  $S$ . Given a hypothesis  $\mathcal{H}$ , a hypothesis verification is used to distinguish true hypotheses from false ones, as described below.



**Fig. 9.** Block diagram of 3D object recognition.



**Fig. 10.** Achieved recognition results on three sample scenes from the UWA dataset. The correctly recognized objects have been superimposed with their complete 3D models from the object database.



**Fig. 11.** Recognition rates on the UWA Dataset.

**Table 4**

Computational efficiency of the proposed technique in comparison with state-of-the-art approaches.

Techniques	Implementation platform	Computation Time (s)
Spin Image	MATLAB	42.2
3D-Tensor	MATLAB	31.8
VD-LSD	MATLAB	20.1
Keypoint/depth	MATLAB	18.6
RoPs	C	7.2
<b>Proposed KSR</b>	<b>MATLAB</b>	<b>3.0</b>

#### 6.4. Hypothesis verification (Module H)

During the hypothesis verification step, a model  $M$  from the library is transformed to the mesh of the scene  $S$  by calculating a rigid transformation for a feature correspondence that votes for the model  $M$ . This is achieved by aligning the coordinate frame  $L_M$  of the model feature at keypoint with the coordinate frame  $L_S$  at keypoint  $p_S$  of the scene feature: where  $R_r$  is the rotation matrix and  $t_r$  is the translation

**Table 3**

Precision and recall values on the Ca'Foscari Venezia dataset.

		Armadillo	Bunny	Cat1	Centaur1	Chef	Chicken	Dog7	Dragon	Face
Precision	KSR	99	100	85	95	100	99	96	96	94
	Game-theoretic	100	100	78	96	93	93	95	100	91
Recall	KSR	99	98	84	98	100	99	88	88	94
	Game-theoretic	97	97	82	100	100	100	86	89	95
Precision		Ganesha	Gorilla0	Horse7	Lioness13	Para	Rhino	T-Rex	Victoria3	Wolf2
	KSR	94	95	96	87	95	92	98	82	80
	Game-theoretic	89	95	97	88	97	91	97	83	82
Recall	KSR	98	90	98	99	93	90	98	89	94
	Game-theoretic	100	91	100	100	94	91	97	83	95



vector of the rigid transformation. It is worth noting that this transformation can be calculated from a single feature correspondence using our proposed KSR technique. This is a major advantage of our fully automatic object recognition algorithm compared to most of the popular techniques e.g., point signature, splash and spin image based methods. These methods require at least three correspondences to calculate a transformation [23].

In the next step, the alignment is further refined using the Iterative Closest Point (ICP) algorithm [41], which results into a residual error  $\varepsilon$ . The candidate model  $M$  and the transformation hypothesis are accepted as being correct only if the residual error  $\varepsilon$  is smaller than a threshold  $\tau_\varepsilon$ . The latter was set empirically to 0.65. If no transformation hypothesis achieves an accurate alignment, the model  $M$  is considered not to be present in the scene  $S$ . In contrast, if more than one hypothesis is accepted, it indicates that multiple instances of the model  $M$  exist in the scene  $S$ .

### 6.5. Recognition and segmentation (Module I)

An accepted hypothesis results in the recognition, pose estimation and segmentation of the library object in the scene. The recognition and segmentation results are then saved. The scene points, which correspond to this model, are subsequently removed from the scene. This not only improves the computational efficiency but also increases the odds for the recognition of the remaining scene objects. After the recognition and the segmentation of an object from the scene, the algorithm proceeds to the next candidate model and the matching process is repeated. This process continues until the complete scene has been segmented or until there are too few points left in the scene.

## 7. 3D object recognition experimental results

To evaluate the performance of the proposed technique, we used KSR to perform 3D object recognition on the UWA [8] and recently proposed Ca'Foscari dataset. UWA dataset contains five 3D models and 50 real scenes. Each scene contains 4 or 5 of the models in the presence of occlusion and clutter.

Ca'Foscari dataset is composed of 20 models and 150 scenes. Each scene contains 3–5 objects in the presence of occlusion and clutter. It is the largest available challenging high resolution 3D object recognition dataset.

Performance evaluation of the proposed technique against existing local features including: Rotation of Project statistics (RoPs) [23], 3D-Tensor [8], VD-LSD [45], keypoint-depth local feature [31], Spherical Harmonics (SH) [46], Spin Images (SI) [25] and Game-theoretic [47] is also presented in this section.

### 7.1. Recognition results on UWA dataset

Fig. 10 (first row) shows three sample scenes from the UWA dataset and their corresponding recognition results are shown in Fig. 10 (second row). Note that all the objects were correctly recognized and segmented from the scene.

We also performed a quantitative evaluation of the proposed technique on the same dataset and compared its performance against state-of-the-art techniques. The recognition rates of the proposed technique are reported in Fig. 11 with respect to varying levels of occlusion. We also report the recognition results of Rotation of Project statistics (RoPs) [23], 3D-Tensor [8], VD-LSD [45], keypoint-depth local feature [31], Spherical Harmonics (SH) [46] and Spin Images (SI) [25].

As shown in Fig. 11, the proposed technique achieves an average recognition rate of 95.9%. The second best results were achieved by 3D-Tensor with an average recognition rate of 94.8%, followed by keypoint based local feature [31], VD-LSD [45], spin images [25] and spherical harmonics [46]. Note that our proposed technique is robust

to occlusion, achieving a 100% recognition rate for up to 75% occlusion. As occlusion further increases, the recognition rate starts to decrease. RoPs achieves the average recognition rate of 98%. This high performance is expected, as the computation of RoPs involves several complex steps. First, projections of the original pointcloud onto three different coordinate planes are obtained. Next, for each projection a 2D bounding rectangle is defined and divided into bins. The number of points falling in each bin are then counted to obtain a matrix. Then, a set of statistics, including central moments and Shannon entropy, are extracted for each projection. The central moments and Shannon entropy collectively form a statistics vector. Three such vectors are concatenated to form a sub feature. Finally, RoPs is constructed by combining all the sub features calculated from the rotation of the pointcloud. On the other hand, our proposed technique is less complex and relies on the computation of the geometrical relationship between the 3D keypoints. In addition to this, the computational efficiency of the proposed KSR is 57% higher than the RoPs descriptor, as demonstrated in the next Section.

### 7.2. Recognition results on Ca'foscari dataset

Table 3 reports the precision and recall values of our proposed technique on Ca'Foscari dataset. The results, as reported in [47] are reproduced here for comparison. Following the procedure in [47], two out of the 20 models were left out from the recognition tests and used as clutter. The proposed technique achieves better precision results for most of the models. The average precision achieved by our proposed technique is 93.5%, while [47] achieves average precision of 92.5%. The average recall of the proposed technique is 94.27%, while [47] achieved 94.20%.

### 7.3. Computational complexity

Table 4 reports the computational efficiency of the proposed technique in comparison with state-of-the-art approaches. These experiments were performed on a 64 bits Intel(R) Corei7, 3.4 GHz, 16 GB RAM. Our code was fully implemented in MATLAB without any parallel computation techniques. Note that the average computation time taken by the proposed technique to compute KSRs is only 3 s, which is much lower than the computation time of the state-of-the-art techniques. This is expected since our approach does not use local surface descriptors. The computational efficiency of the proposed technique is expected to improve many folds once the algorithm is implemented in C++, which will make it suitable for real-time applications. The second best is achieved by RoPs followed by keypoint-depth features. These results suggest that the proposed technique achieves the best computational efficiency compared to state-of-the-art methods.

## 8. Conclusion

In this paper, we presented a novel technique which is capable of capturing the geometrical relationship between 3D keypoints. In contrast to existing methods, the proposed technique does not rely on the extraction of local surface features around detected keypoints, thus omitting this computationally expensive step. It is also robust to variations in the noise levels and mesh resolutions as demonstrated by experimental results on the Bologna dataset. The proposed technique has been extensively tested for the tasks of 3D modeling and 3D object recognition. It achieves highly accurate 3D modeling on the UWA and Stanford 3D models dataset. It also achieves good 3D object recognition results and outperforms the state-of-the-art techniques by achieving a high recognition rate of 95.9% on UWA and 93.5% on Ca'Foscari dataset, and an average computation time of only 3 s.

## Acknowledgments

This research is supported by the University of Western Australia (UWA) and Australian Research Council (ARC) grant DP110102166.

## References

- [1] A. Mademlis, P. Daras, D. Tzovaras, M.G. Strintzis, 3d object retrieval using the 3d shape impact descriptor, *Pattern Recognit.* 42 (2009) 2447–2459.
- [2] M. Hasegawa, S. Tabbone, Amplitude-only log radon transform for geometric invariant shape descriptor, *Pattern Recognit.* 47 (2014) 643–658.
- [3] S.A.A. Shah, M. Bennamoun, F. Boussaid, A novel 3D vorticity based approach for automatic registration of low resolution range images, *Pattern Recognit.* 48 (2015) 2859–2871.
- [4] J. Novatnack, K. Nishino, *Computer Vision – ECCV 2008 in: Proceedings of the 10th European Conference on Computer Vision, Marseille, France, October 12–18, 2008, Proceedings, Part III, Springer Berlin Heidelberg, Berlin, Heidelberg*, pp. 440–453.
- [5] D.H. Chung, I.D. Yun, S.U. Lee, Registration of multiple-range views using the reverse-calibration technique, *Pattern Recognit.* 31 (1998) 457–464.
- [6] Y. Gao, Q. Dai, N.-Y. Zhang, 3d model comparison using spatial structure circular descriptor, *Pattern Recognit.* 43 (2010) 1142–1151.
- [7] S.A.A. Shah, M. Bennamoun, F. Boussaid, A novel feature representation for automatic 3d object recognition in cluttered scenes, *Neurocomputing* 205 (2016) 1–15.
- [8] A.S. Mian, M. Bennamoun, R. Owens, Three-dimensional model-based object recognition and segmentation in cluttered scenes, *IEEE Trans. Pattern Anal. Mach. Intell.* 28 (2006) 1584–1601.
- [9] X. Song, D. Muselet, A. Trémeau, Affine transforms between image space and color space for invariant local descriptors, *Pattern Recognit.* 46 (2013) 2376–2389.
- [10] S.A.A. Shah, M. Bennamoun, F. Boussaid, A. A. El-Sallam, On novel local surface description for automatic object recognition in low resolution cluttered scenes, in: *International Conference on Computer Vision (ICCV) workshop*.
- [11] E. Bayro-Corrochano, Motor algebra approach for visually guided robotics, *Pattern Recognit.* 35 (2002) 279–294.
- [12] B. Barshan, T. Aytac, C. Yüzbaşıoğlu, Target differentiation with simple infrared sensors using statistical pattern recognition techniques, *Pattern Recognit.* 40 (2007) 2607–2620.
- [13] S.A.A. Shah, M. Bennamoun, F. Boussaid, Iterative deep learning for image set based face and object recognition, *Neurocomputing* 174 (2016) 866–874.
- [14] J. Williams, M. Bennamoun, A multiple view 3d registration algorithm with statistical error modeling, *IEICE Trans. Inf. Syst.* 83 (2000) 1662–1670.
- [15] W. Fu, J. Wang, H. Lu, S. Ma, Dynamic scene understanding by improved sparse topical coding, *Pattern Recognit.* 46 (2013) 1841–1850.
- [16] C. Shi, C. Wang, B. Xiao, S. Gao, J. Hu, End-to-end scene text recognition using tree-structured models, *Pattern Recognit.* 47 (2014) 2853–2866.
- [17] M.R. Ogiela, R. Tadeusiewicz, Artificial intelligence structural imaging techniques in visual pattern analysis and medical data understanding, *Pattern Recognit.* 36 (2003) 2441–2452.
- [18] S.A.A. Shah, K. Yahya, G. Mubashar, A. Bais, Quantification and visualization of mri cartilage of the knee: A simplified approach, in: *Emerging Technologies (ICET), 2010 Proceedings of the 6th International Conference on, IEEE*, pp. 175–180.
- [19] E. Sesa-Nogueras, M. Faundez-Zanuy, Biometric recognition using online upper-case handwritten text, *Pattern Recognit.* 45 (2012) 128–144.
- [20] S.H. Khan, M.A. Akbar, F. Shahzad, M. Farooq, Z. Khan, Secure biometric template generation for multi-factor authentication, *Pattern Recognit.* 48 (2015) 458–472.
- [21] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, 3d object recognition in cluttered scenes with local surface features: a survey, *IEEE Trans. Pattern Anal. Mach. Intell.* 36 (2014) 2270–2287.
- [22] C. Geng, X. Jiang, Face recognition based on the multi-scale local image structures, *Pattern Recognit.* 44 (2011) 2565–2575.
- [23] Y. Guo, F. Sohel, M. Bennamoun, M. Lu, J. Wan, Rotational projection statistics for 3d local surface description and object recognition, *Int. J. Comput. Vis.* 105 (2013) 63–86.
- [24] R. Osada, T. Funkhouser, B. Chazelle, D. Dobkin, Shape distributions, *ACM Trans. Graph.* 21 (2002) 807–832.
- [25] A. Johnson, M. Hebert, Using spin images for efficient object recognition in cluttered 3d scenes, *IEEE Trans. Pattern Anal. Mach. Intell.* 21 (1999) 433–449.
- [26] B. Taati, M. Bondy, P. Jasiobedzki, M. Greenspan, Variable dimensional local shape descriptors for object recognition in range data, in: *Computer Vision, 2007. ICCV 2007. IEEE in: Proceedings of the 11th International Conference on, IEEE*, pp. 1–8.
- [27] F. Tombari, S. Salti, L. Stefano, Unique signatures of histograms for local surface description, in: K. Daniilidis, P. Maragos, N. Paragios (Eds.), *Computer Vision ECCV 2010, volume 6313 of Lecture Notes in Computer Science, Springer Berlin Heidelberg*, 2010, pp. 356–369.
- [28] Y. Keller, T. Darom, Scale invariant features for 3d mesh models, *IEEE Trans. Image Process.* 21 (2012) 2758–2769.
- [29] C. Li, J. Li, D. Gao, B. Fu, Rapid-transform based rotation invariant descriptor for texture classification under non-ideal conditions, *Pattern Recognit.* 47 (2014) 313–325.
- [30] I. Sipiran, B. Bustos, Harris 3d: a robust extension of the harris operator for interest point detection on 3d meshes, *Vis. Comput.* 27 (2011) 963–976.
- [31] A. Mian, M. Bennamoun, R. Owens, On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes, *Int. J. Comput. Vis.* 89 (2010) 348–361.
- [32] H. Chen, B. Bhanu, 3d free-form object recognition in range images using local surface patches, *Pattern Recognit. Lett.* 28 (2007) 1252–1262.
- [33] E. Moors, On the reciprocal of the general algebraic matrix, abstract, *Bull. Am. Math. Soc.* 26 (1920) 394–395.
- [34] R. Penrose, A generalized inverse for matrices, in: *Proceedings Cambridge Philos. Soc.*, volume 51, Cambridge Univ Press, pp. 406–413.
- [35] R. Unnikrishnan, M. Hebert, Multi-scale interest regions from unorganized point clouds, in: *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on, IEEE*, pp. 1–8.
- [36] H. Chen, B. Bhanu, 3d free-form object recognition in range images using local surface patches, *Pattern Recognit. Lett.* 28 (2007) 1252–1262.
- [37] Y. Zhong, Intrinsic shape signatures: A shape descriptor for 3d object recognition, in: *Computer Vision Workshops (ICCV Workshops), 2009 IEEE Proceedings of the 12th International Conference on*, pp. 689–696.
- [38] J. Sun, M. Ovsjanikov, L. Guibas, A concise and provably informative multi-scale signature based on heat diffusion, in: *Computer Graphics Forum*, volume 28, Wiley Online Library, pp. 1383–1392.
- [39] T. Hou, H. Qin, Efficient computation of scale-space features for deformable shape correspondences, in: *Computer Vision—ECCV 2010, Springer*, 2010, pp. 384–397.
- [40] A. Flint, A. Dick, A. Van den Hengel, Local 3d structure recognition in range images, *IET Comput. Vis.* 2 (2008) 208–217.
- [41] P.J. Besl, N.D. McKay, Method for registration of 3-d shapes, in: *Robotics-DL tentative, International Society for Optics and Photonics*, pp. 586–606.
- [42] The Stanford 3d scanning repository, (<http://graphics.stanford.edu/data/3Dscanrep/>), 2003.
- [43] P.J. Neugebauer, Reconstruction of real-world objects via simultaneous registration and robust combination of multiple range images, *Int. J. Shape Model.* 3 (1997) 71–90.
- [44] H.Q. Dinh, S. Kropac, Multi-resolution spin-images, in: *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, IEEE, pp. 863–870.
- [45] B. Taati, M. Greenspan, Local shape descriptor selection for object recognition in range data, *Comput. Vis. Image Underst.* 115 (2011) 681–694.
- [46] M. Kazhdan, T. Funkhouser, S. Rusinkiewicz, Rotation invariant spherical harmonic representation of 3d shape descriptors, in: *Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, SGP '03, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 2003*, pp. 156–164.
- [47] E. Rodolà, A. Albarelli, F. Bergamasco, A. Torsello, A scale independent selection process for 3d object recognition in cluttered scenes, *Int. J. Comput. Vis.* (2013) 1–17.

**Syed Afaq Ali Shah** obtained his PhD from the University of Western Australia in the area of computer vision and machine learning. He is currently working as a research associate in school of computer science and software engineering, the University of Western Australia, Crawley, Australia. His research interests include 3D object recognition, 3D modeling, deep learning and image processing.

**M. Bennamoun** received his M.Sc. degree in control theory from Queen's University, Kingston, ON, Canada, and the Ph.D. degree in computer vision from Queensland University of Technology (QUT), Brisbane, Australia. He lectured Robotics at Queen's University and then joined QUT in 1993 as an Associate Lecturer. He is currently a Winthrop Professor and has been the Head of the School of Computer Science and Software Engineering, The University of Western Australia (UWA), Perth, Australia for five years (Feb, 2007–Feb, 2012). He has published over 200 journal and conference publications and secured highly competitive national grants from the Australian Research Council (ARC). His areas of interest include control theory, robotics, obstacle avoidance, object recognition, artificial neural networks, signal/image processing, and computer vision (particularly 3D).

**F. Boussaid** received the M.S. and Ph.D. degrees in microelectronics from the National Institute of Applied Science (INSA), Toulouse, France, in 1996 and 1999, respectively. He joined Edith Cowan University, Perth, Australia, as a Postdoctoral Research Fellow, and a member of the Visual Information Processing Research Group in 2000. He joined the University of Western Australia, Crawley, Australia, in 2005, where he is currently a Professor. His current research interests include smart CMOS vision sensors and image processing.