Auton Robot
DOI 10.1007/s10514-016-9612-y

B-SHOT: a binary 3D feature descriptor for fast Keypoint matching on 3D point clouds

Sai Manoj Prakhya[1] · **Bingbing Liu**[2] · **Weisi Lin**[1] · **Vinit Jakhetiya**[3] · **Sharath Chandra Guntuku**[1]

Received: 15 February 2016 / Accepted: 16 December 2016

Abstract We present the first attempt in creating a binary 3D feature descriptor for fast and efficient keypoint matching on 3D point clouds. Specifically, we propose a binarization technique and apply it on the state-of-the-art 3D feature descriptor, SHOT (Salti et al., Comput Vision Image Underst 125:251–264, 2014) to create the first binary 3D feature descriptor, which we call *B-SHOT*. B-SHOT requires 32 times lesser memory for its representation while being six times faster in feature descriptor matching, when compared to the SHOT feature descriptor. Next, we propose a robust evaluation metric, specifically for 3D feature descriptors. A comprehensive evaluation on standard benchmarks reveals that B-SHOT offers comparable keypoint matching performance to that of the state-of-the-art real valued 3D feature descriptors, albeit at dramatically lower computational and memory costs.

✉ Bingbing Liu
bliu@i2r.a-star.edu.sg

Sai Manoj Prakhya
saimanoj001@ntu.edu.sg

Weisi Lin
wslin@ntu.edu.sg

Vinit Jakhetiya
vjakhetiya@ust.hk

Sharath Chandra Guntuku
sharathc001@ntu.edu.sg

[1] School of Computer Engineering, Nanyang Technological University, Singapore, Singapore

[2] Institute for InfoComm Research, A*STAR, Singapore, Singapore

[3] Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong, Hong Kong

Keywords 3D Binary feature descriptors · 3D keypoint matching · Binarization · Point cloud registration

1 Introduction

Efficient keypoint matching is a pre-requisite for various applications such as 3D object recognition (Guo et al. 2014a; Aldoma et al. 2012), simultaneous localization and mapping (SLAM) (Endres et al. 2012), Sparse Depth Odometry (SDO) (Prakhya et al. 2015b), 3D shape retrieval and range image/point cloud registration (Guo et al. 2014b). All these applications involve keypoint detection and their matching via feature descriptors to find true keypoint correspondences. There are various 3D keypoint detectors (Rusu et al. 2011; Fiolka et al. 2012; Steder et al. 2011; Zhong 2009; Rodolà et al. 2015; Prakhya et al. 2016) available in literature to detect salient/interest points on 3D point clouds.[1] Once the keypoints are detected on 3D point clouds, the next step is to match them via feature descriptors. Feature descriptors essentially encode the information in the neighbourhood of a keypoint into a multi-dimensional vector. The feature descriptors are generally matched by calculating Euclidean distance in their high dimensional vector spaces and this is a computationally expensive step. Moreover, the computational requirements for feature descriptor matching and the memory footprint increases proportionally with the increase in the dimensionality of the feature descriptor.

To develop mobile applications that require less memory and lower computational power, binary feature descriptors (Calonder et al. 2012; Tra et al. 2015; Alahi et al. 2012; Leutenegger et al. 2011; Strecha et al. 2012; Trzcinski et al.

[1] Tombari et al. (2013) presented a comprehensive survey and performance evaluation of various 3D keypoint detectors.

Published online: 30 December 2016

2015) have been heavily researched and developed in 2D image domain. They are extensively used in various applications such as 2D object detection/tracking, image retrieval (Zhou et al. 2015), visual odometry (Leutenegger et al. 2015) and place recognition (Galvez-Lopez and Tardos 2012) in SLAM for robotics. Binary feature descriptors win over traditional real valued feature descriptors as they can be matched extremely fast via Hamming distance metric and have dramatically less memory footprint.

With the advent of various handheld depth sensors such as Microsoft Kinect, Asus Xtion Pro Live, Structure sensor, Intel RealSense camera and Google Tango, 3D data acquisition has become affordable. Since then, there has been a surge of consumer applications that employ 3D sensors and process dense depth data on mobile devices for various vision and robotic applications. A recent project of Google's ATAP, Project Tango, has received a lot of attention as it can provide online 3D pose estimates and dense depth data from a mobile device. With the availability of these mobile 3D data acquisition devices, there is a need to develop applications that have a low memory footprint and require less computational power. Moreover, with the development of hand-held 3D scanning algorithms, such as KinectFusion (Newcombe et al. 2011) and CopyMe3D (Sturm et al. 2013; Choi et al. 2015), various applications such as, 3D object recognition (Guo et al. 2014a; Aldoma et al. 2012) and 3D shape retrieval (Tabia et al. 2014) are gaining a lot of research attention. These factors necessitate the development of binary feature descriptors in 3D domain as well to develop applications for memory and power constrained devices.

Apart from reducing memory footprint and computational costs, a 3D binary feature descriptor can be used to aid 2D feature descriptor matching to achieve much higher accuracy. Though there are quite a few binary feature descriptors in the 2D image domain, there are no binary feature descriptors in the 3D point cloud domain. There are three main reasons for this. Firstly, there is no attribute in 3D point cloud data, $[x\ \ y\ \ z]$, that is equivalent to the pixel intensity in 2D image domain. Secondly, 3D data from affordable sensors like Microsoft Kinect is quite noisy, hence simple thresholding, which is often used in 2D binary feature descriptors, turns out to be ineffective. Lastly, the 3D point cloud data can be unordered, i.e., neighbourhood connectivity information may not be present, whereas 2D images are ordered.

Contribution: We propose a new binarization technique and apply it on a state-of-the-art 3D feature descriptor, SHOT, to create its binary counterpart, B-SHOT. The advantages of B-SHOT over the traditional SHOT feature descriptor are as follows. Firstly, SHOT, which is of 352 dimensions (requiring 1408 bytes) is reduced to 352 bits of binary data, therefore reducing the memory footprint, by 32 times. Secondly, keypoint matching can be performed much faster with Hamming distance metric when compared to real valued 3D feature descriptors (Experiments show a 6× improvement). Then, we propose a robust evaluation metric for 3D feature descriptor evaluation that considers 3D keypoint detection ambiguity and all possible false feature correspondences. A comprehensive evaluation on benchmark datasets shows that B-SHOT offers comparable keypoint matching performance to that of the state-of-the-art 3D feature descriptors with added advantages of having low memory and computational requirements.

Paper organization: We generalize and extend our earlier work (Prakhya et al. 2015a) in this paper. In Sect. 2, we first present a brief overview of the state-of-the-art real valued 3D feature descriptors and highlight the limitations in extending the techniques used in 2D binary feature descriptors to 3D domain. We thus propose a binarization technique to convert a real valued 3D feature descriptor to binary descriptor in Sect. 3. In Sect. 4, we perform extensive evaluation of B-SHOT on publicly available datasets. The proposed binarization has two free parameters, hence in Sect. 5, we test the robustness of B-SHOT to those parameter variations. Lastly, in Sect. 6, we present the computational and memory requirements of the proposed algorithm and conclude this work.

2 Related work

Here, we present the state-of-the-art real valued 3D feature descriptors as well as a brief overview of 2D binary feature descriptors.

2.1 Real valued 3D feature descriptors

Salti et al. (2014) have classified the existing 3D feature descriptors into two classes, namely signature based methods and histogram based methods. Signature based feature descriptors (Chua and Jarvis 1997; Darom and Keller 2012; Novatnack et al. 2008; Mian et al. 2010; Knopp et al. 2010) that encode relative spatial information from the neighbourhood points are highly descriptive. However, they are not robust because even small variations in defining the local reference frame can make two descriptors substantially different. On the other hand, histogram based methods (Johnson and Hebert 1999; Rusu et al. 2008, 2009; Chen and Bhanu 2007; Frome et al. 2004; Marton et al. 2010; Zhong 2009; Tombari et al. 2010a) that encode the count of various geometrical properties in the considered support region are robust to small changes in 3D point positions but lack the property of being highly descriptive as they do not encode relative spatial information. Hence, Salti et al. (2010b, 2011, 2014) have proposed SHOT feature descriptor, as explained in Sect. 2.1.1, which has the advantages of both signature and histogram based methods. Apart from these, a game theory based technique (Albarelli et al. 2010a, b) with global geometric consistency was proposed for fine surface matching/registration and was successfully applied to 3D object recognition (Rodolà et al. 2012) as well.

Recently, Guo et al. (2015) have performed an extensive performance evaluation of 3D local feature descriptors and highlighted that SHOT (Salti et al. 2014), RoPS (Guo et al. 2013) and FPFH (Rusu et al. 2009) feature descriptors offer consistently good performance in various scenarios while having some trade-off in terms of descriptiveness, computational power and storage requirements. Our proposed method to convert a real valued feature descriptor offers good performance when applied to SHOT, rather than RoPS or FPFH feature descriptors. The reasons for this are discussed later and are fortified with experimental results. As we use SHOT (Salti et al. 2014) feature descriptor extensively, we present a detailed explanation of its working in Sect. 2.1.1 while briefly discussing RoPS and FPFH feature descriptors in this section.

Rotational projection statistics (RoPS) (Guo et al. 2013) is a hybrid method that has advantages of both signature and histogram based methods (Salti et al. 2014). RoPS capitalizes on creating a robust local reference frame based on the eigenvalue decomposition of the neighbourhood covariance matrix. RoPS employs small modifications as it works on meshes, while SHOT works on raw point clouds. Similar to all the descriptors that employ a local reference frame, invariance to rigid transformations is achieved by aligning the local surface patch to the estimated local reference frame. Then the local surface patch is rotated around the x, y, z axes while projecting them onto xy, yz, xz planes. From each projection, a distribution matrix is constructed and its statistics, such as central moments for descriptiveness and Shannon entropy to capture the information in a probability distribution, are calculated. Subsequently, these statistics are concatenated to create a 135 dimensional RoPS feature descriptor.

Fast point feature histograms (FPFH) (Rusu et al. 2009) descriptor has gained a lot of popularity recently through point cloud library (Rusu et al. 2011). FPFH is an enhancement to the initially proposed point feature histogram (PFH) (Rusu et al. 2008), wherein a Darboux frame is created at every point in the local neighbourhood. Then, three angles between each pair of points and the distance between them is calculated and binned to create a 125 dimensional feature descriptor. In FPFH, which is an enhanced version of PFH, the distance parameter is dropped as it does not cater to point cloud density variations. Moreover, FPFH is sped up by calculating Simple PFH and weighing them to create a 33 dimensional FPFH feature descriptor.

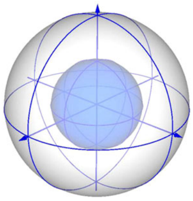


Fig. 1 Spherical Grid employed in SHOT (Salti et al. 2014) descriptor

2.1.1 SHOT feature descriptor

In our proposed method, we transform the SHOT (Salti et al. 2014) feature descriptor into a binary representation, B-SHOT. Hence for completeness of the paper, we describe the working principles behind the SHOT feature descriptor in this section. SHOT is one of the state-of-the-art 3D feature descriptors as shown by extensive comparative evaluations (Guo et al. 2015; Salti et al. 2014; Tombari et al. 2010b) along with other prominent 3D feature descriptors such as ROPS (Guo et al. 2013) and FPFH (Rusu et al. 2009). Inspired by SIFT (Lowe 2004) and considering the advantages of both signature and histogram based methods, (Salti et al. 2014) have developed Signatures of Histograms of OrienTations (SHOT) descriptor.

In order to make the feature descriptor invariant to rotation and translation, the authors propose to create a local reference frame from the eigen vector of the modified covariance matrix $\mathbf{C}$, as shown in Eq. 1. The authors weigh the sample points $\mathbf{q_i}$ that lie in the support region of radius $\mathbf{r}$ based on their distance from the considered point $\mathbf{q}$,

$$\mathbf{C} = \frac{1}{\sum_{i:d_i \leq \mathbf{r}} (\mathbf{r} - d_i)} \sum_{i:d_i \leq \mathbf{r}} (\mathbf{r} - d_i)(\mathbf{q_i} - \mathbf{q})(\mathbf{q_i} - \mathbf{q})^{\mathbf{T}} \qquad (1)$$

where $d_i = ||q_i - q||_2$. To create a unique local reference frame and remove the sign ambiguity, the direction of the local $\mathbf{x}$ and $\mathbf{z}$ axes are oriented towards the majority direction of the vectors that they represent. Finally, the local $\mathbf{y}$ axis is obtained by the cross product of $\mathbf{z}$ and $\mathbf{x}$, i.e., $\mathbf{y} = \mathbf{z} \times \mathbf{x}$. The local surface patch is aligned with this estimated local reference frame to achieve invariance to rigid transformations.

To create a signature like structure, a 3D isotropic spherical grid is aligned with the estimated local reference frame. This 3D spherical grid has 32 partitions arising from 8 azimuth, 2 elevation and 2 radial divisions, as shown in Fig. 1 (note that only 4 azimuth partitions are shown for better visibility). The 3D point distribution in each of these 32 partitions is represented by a local histogram created by binning the cosine of the angle between the $\mathbf{z}$ axis at feature point $\mathbf{q}$ and the normals at the neighbourhood points $\mathbf{q_i}$ that lie in the support region. Uniform binning on the cosine angle is equivalent to applying a coarse binning near the local $\mathbf{z}$ axis and a finer one at the orthogonal direction in the spatial domain, hence making it robust to small variations in surface normals. To cope with the boundary effects arising from histogram based binning and small perturbations in the local reference frame, a quadrilinear interpolation technique is employed. Finally the descriptor is normalized to make it robust against point density variations.

2.2 2D binary feature descriptors

Coming to the binary feature descriptors in the 2D image domain, most of them (Calonder et al. 2012; Leutenegger et al. 2011; Alahi et al. 2012) are constructed by comparing the intensity values of pixels that lie in the neighbourhood/support region. They vary in terms of the sampling pattern employed for comparison. BRIEF (Calonder et al. 2012) compares the intensity values of the center pixel with another being chosen from an isotropic Gaussian distribution. FREAK (Alahi et al. 2012) and BRISK (Leutenegger et al. 2011) use sampling patterns of overlapping concentric circles and differ in the number of points that lie in inner and outer rings. It is not straightforward to extend the pixel based thresholding methods (Calonder et al. 2012; Alahi et al. 2012; Leutenegger et al. 2011) employed in the 2D image domain to 3D point clouds as there is no attribute in 3D point clouds that is equivalent to pixel intensity in 2D images and the 3D point clouds can be unordered. There are other works that employ hashing (Strecha et al. 2012), learning based approaches (Trzcinski et al. 2015; Yang and Cheng 2014; Gong et al. 2013) to create 2D binary feature descriptors as well. However, the lack of standard and publicly available 3D data for training purposes, the difference that arises from 2.5D and 3D depth data and point cloud density variations make it difficult to import these learning based techniques. Hence, we propose a binarization method that is data-independent and do not require any learning, to convert a real valued 3D feature descriptor into a binary descriptor for fast keypoint matching on 3D point clouds.

There are other class of works in 2D image domain which binarize the real valued 2D feature descriptors, such as SIFT (Lowe 2004), to create a binary feature descriptor. B-SIFT (Zhou et al. 2015) calculates the median of a considered SIFT feature descriptor, assigns a binary value of ‘1’ if the descriptor value is greater than the calculated median and ‘0’ otherwise, which results in a 128 bit binary SIFT descriptor. They also extend it to a 256 bit descriptor by thresholding with two medians taken from the considered SIFT feature descriptor. Dominant-SIFT (Tra et al. 2015) uses the binary coded position of the cumulative maximum value in each sub-histogram of the SIFT feature descriptor to create a 48-bit binary feature descriptor.

We take inspiration from B-SIFT (Zhou et al. 2015) and Dominant-SIFT (Tra et al. 2015) algorithms in proposing our binarization technique to convert a real valued feature descriptor to a binary feature descriptor. Our experiments showed that the Dominant—SIFT’s binarization technique performed very poorly on SHOT feature descriptor. There are two reasons, firstly, the encoded binary position of the maximum value in each sub-histogram can vary significantly in noisy 3D data (Guo et al. 2015; Salti et al. 2014). Secondly, many feature descriptors can have exactly same binary representation (in the case of Dominant-SIFT) (Tra et al. 2015), resulting in many false correspondences. Our proposed binarization scheme is adaptive and stores more information thereby resulting in better performance when compared to B-SIFT’s quantization scheme. Moreover, when we applied our binarization technique to SIFT in the 2D domain, it offered superior performance to Dominant SIFT (Tra et al. 2015) and B-SIFT (Zhou et al. 2015) on benchmarks proposed by Mikolajczyk et al. (2005). However, in this paper, we confine ourselves to evaluating the proposed binarization technique comprehensively on 3D feature descriptors alone.

There is only one notable work (Malaguti et al. 2012) in 3D domain that attempts to compress SHOT feature descriptor. This work is commendable as it can compress a 352 dimensional SHOT descriptor to 528 bits while having a minimal drop in performance, however, it has major shortcomings. First, the SHOT feature descriptor has to be compressed on the transmitter side, then should be decompressed on the receiver side and finally is matched using Euclidean distance metric. Second, this work does not hold any significance for on-device applications as the feature descriptor matching is still Euclidean metric based, whereas for remote applications that require online transfer of 3D feature descriptors, there is an overhead of computations for both decompression and Euclidean metric based matching while having the same memory footprint on the receiver side as that of the 352 dimensional SHOT feature descriptor. In our proposal, there is no need for decompression and the binary descriptors are directly matched using Hamming distance metric, which significantly reduces the computational time, when compared to Euclidean metric based matching, as shown later in Sect. 6. Though our proposal shows a slight drop in performance, it has dramatically less memory (32×) and computational requirements (6×) while having relevance to both on-device applications, as the matching is faster than traditional feature descriptor matching, and remote applications, as only 352 bits needs to be transmitted without any need for decompression, enables faster matching and has a memory footprint of only 352 bits on the receiver side. Finally, our work opens a new research direction to create even more efficient binary 3D feature descriptors.

3 The proposed technique to binarize SHOT

The SHOT feature descriptor is a 352 dimensional vector[2] comprising of 11 bin histograms arising from 32 spatial grids in 3D space. Each histogram represents the angles that the surface normals in a certain spatial grid make with the local reference frame at the considered keypoint. To create a B-SHOT feature descriptor, we encode the 352 dimensional SHOT descriptor into a 352 bit binary descriptor. Let us consider a SHOT descriptor S_i, where $i = \{0, 1, 2, \ldots, 351\}$ and each value of S_i can be any decimal value between 0 and 1. Let us represent the newly created B-SHOT feature descriptor by B_i, where $i = \{0, 1, 2, \ldots, 351\}$ and each value of B_i is either 0 or 1. This binarization of SHOT to B-SHOT is performed in an iterative procedure that takes 'm' consecutive values from the beginning of S_i and encodes them into corresponding m binary bits in B_i. In the proposed binarization technique, we roughly preserve the shape of the sub-histograms from various spatial grids that are concatenated to form a SHOT feature descriptor.

As mentioned above, 'm' represents the number of real values that are encoded into binary bits. For the case where $m = 4$, consider four values $\{S_0, S_1, S_2, S_3\}$ from the SHOT feature vector, S_i, and its corresponding four bits $\{B_0, B_1, B_2, B_3\}$ to be encoded into B-SHOT descriptor, B_i.

Following are the various possibilities of encoding S_i into B_i:

Let $S_{sum} = S_0 + S_1 + S_2 + S_3$.

- *Case A*: If all four values in S_i are zeros, then the corresponding four bits of B_i are also set to zero, i.e., $\{B_0, B_1, B_2, B_3\}$ will be $\{0, 0, 0, 0\}$ in this case.
- *Case B*: If Case A does not hold, we check if there is a single value, S_i, $S_i \in \{S_0, S_1, S_2, S_3\}$ that amounts to 90% of S_{sum}. If yes, then its position is coded in a binary fashion. For example, if S_1's value amounts to more than 90% of the S_{sum}, then the encoded $\{B_0, B_1, B_2, B_3\} = \{0, 1, 0, 0\}$. In this way, four cases are covered and the possible values of $\{B_0, B_1, B_2, B_3\}$ are $\{1, 0, 0, 0\}$, $\{0, 1, 0, 0\}$, $\{0, 0, 1, 0\}$, $\{0, 0, 0, 1\}$.
- *Case C*: If Case A and Case B do not hold, then we check if the sum of any two values amount to 90% of S_{sum}. For example, if the sum of S_0 and S_3 amounts to more than 90% of S_{sum}, then the encoded $\{B_0, B_1, B_2, B_3\} = \{1, 0, 0, 1\}$. In this way, the possible values of $\{B_0, B_1, B_2, B_3\}$ are $\{1, 1, 0, 0\}$, $\{1, 0, 1, 0\}$, $\{1, 0, 0, 1\}$, $\{0, 1, 1, 0\}$, $\{0, 1, 0, 1\}$ and $\{0, 0, 1, 1\}$. Pseudocode for this case is shown in Algorithm 1.

```
if {!Case A} and {!Case B}
then
  if $\{S_0 + S_1\} > 0.9 \times S_{sum}$
    then   $\{B_0, B_1, B_2, B_3\} = \{1, 1, 0, 0\}$
  else if $\{S_0 + S_2\} > 0.9 \times S_{sum}$
    then   $\{B_0, B_1, B_2, B_3\} = \{1, 0, 1, 0\}$
  else if $\{S_0 + S_3\} > 0.9 \times S_{sum}$
    then   $\{B_0, B_1, B_2, B_3\} = \{1, 0, 0, 1\}$
  else if $\{S_1 + S_2\} > 0.9 \times S_{sum}$
    then   $\{B_0, B_1, B_2, B_3\} = \{0, 1, 1, 0\}$
  else if $\{S_1 + S_3\} > 0.9 \times S_{sum}$
    then   $\{B_0, B_1, B_2, B_3\} = \{0, 1, 0, 1\}$
  else if $\{S_2 + S_3\} > 0.9 \times S_{sum}$
    then   $\{B_0, B_1, B_2, B_3\} = \{0, 0, 1, 1\}$
  end if
end if
```

Algorithm 1: Pseudocode for Case C

- *Case D*: If Case A, Case B and Case C do not hold, then we check if the sum of any three values amounts to more than 90% of S_{sum}. The possible values of $\{B_0, B_1, B_2, B_3\}$ in this case turn out to be $\{1, 1, 1, 0\}$, $\{0, 1, 1, 1\}$, $\{1, 1, 0, 1\}$ and $\{1, 0, 1, 1\}$.
- *Case E*: If none of the above conditions hold, then it means that $\{S_0, S_1, S_2, S_3\}$ are nearly the same values and the encoded $\{B_0, B_1, B_2, B_3\}$ would be $\{1, 1, 1, 1\}$.

We apply the proposed binarization technique on all consecutive m sets in a 352 dimensional SHOT descriptor, i.e., $\{S_0, S_1, S_2, S_3\}$ is encoded as $\{B_0, B_1, B_2, B_3\}$, $\{S_4, S_5, S_6, S_7\}$ is encoded as $\{B_4, B_5, B_6, B_7\}$, and so on, which finally results in a 352 bit binary descriptor, B-SHOT.

In the proposed binarization technique, there are two free variables, m (number of values that are being encoded into binary bits) and E_r (encoding ratio which was set to 0.9 or 90% in the above scenario). Depending on the value of m, there will be 2^m conditions to check for, while performing the binarization. This evaluation of 2^m conditions can be easily simplified by employing a sorting algorithm that sorts the m input real values and then encodes the indices of the minimal set of sorted values that amount to $E_r \times S_{sum}$. We present the pseudocode for this in Algorithm 2. We take m real values as the input and create an array, *vector_pair*, in which each element is $\{S_i, i\}$. The array, *vector_pair* is sorted only based on the values of S_i. Then we iterate through *vector_pair* and find the first j elements that amount to 90% of the S_{sum}. To create the binary vector of m values, the corresponding indices of the j elements are set to 1 while others remain 0.

The proposed binarization method was developed based on the intuitions gained from the structure and the distribution of SHOT descriptors on various 3D point clouds. The following observations were made from various experiments on different 3D point clouds, for example, let us consider an indoor scene, a 3D point cloud as shown in the project website,[3] and extract 1000 SHOT descriptors from uniform

[2] We employ the default implementation of SHOT feature descriptor available through Point Cloud Library at www.pointclouds.org.

[3] https://sites.google.com/site/bshotdescriptor/scene-for-shot.

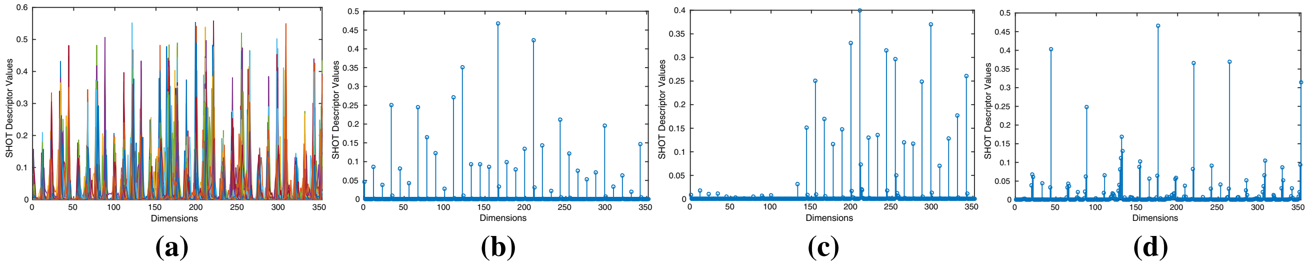


Fig. 2 **a** illustrates the distribution of 1000 SHOT feature descriptors from a scene. **b**, **c** and **d** shows three randomly picked SHOT descriptors from the scene

keypoints on it. Figure 2a shows the distribution of these 1000 SHOT descriptors. It can be seen from Fig. 2a that there exists a repetitive pattern in SHOT descriptor's values, and this was the reason behind choosing an iterative method to binarize a chunk of real valued SHOT descriptor to a binary B-SHOT. Next, Figs. 2b, c and d show three randomly picked SHOT descriptors from the same scene. It can be observed from these figures that there can be peak values in SHOT descriptor with neighbourhood bins going to nearly zero or with nearly same neighbourhood values or they can decrease gradually.

To accommodate all the possible scenarios, the proposed binarization method had Case B, which catered for peaked values, Case C & Case D, which catered for gradual decrease in the histogram neighbourhood values and lastly Case E that checked if all the neighbourhood bins are similar. However, still the chunk size m and the encoding ratio E_r, which catered for the peaked value in the chunk are yet to be determined. Hence, we perform a parameter evaluation experiment by varying these two free parameters, m and E_r, and show the performance of B-SHOT in Sect. 5. It was observed from the experiments in Sect. 5 that B-SHOT is robust to these two parameter variations, m and E_r, hence making it reliable. However, with the values of $m = 4$ and $E_r = 0.9$, there is a slight better performance and the implementation boils down to a set of 16 *'if-else'* conditions. Hence we use these settings while evaluating the performance of B-SHOT.

It is important to note that there is loss of information while converting a SHOT feature descriptor into a B-SHOT feature descriptor. Mainly, in Case C and Case D, the individual contributions made by each of the four values of S_i are ignored if they sum to 90% of S_{sum}. For example, let us look at this extreme case, if $\{S_0, S_1, S_2, S_3\} = \{0.65, 0.20, 0, 0\}$, then the encoded $\{B_0, B_1, B_2, B_3\}$ would be $\{1, 1, 0, 0\}$. As can be seen from the above example, the binary representation highlights that bits B_0 and B_1 are the same, but in reality they are not. It should be noted that, as a result of information loss and quantization effects in the proposed binarization technique, a slightly lower number of correspondences are found by B-SHOT when compared to those found by SHOT.

```
Input: $\{S_0, S_1, \dots, S_{m-1}\}$ real values, $E_r$, $m$
Output: Bit = $\{B_0, B_1, \dots, B_{m-1}\}$ binary values.
Initialization: Create a $vector\_pair$ with
$S_{values}$ : $\{S_0, S_1, \dots, S_{m-1}\}$ and their index values as
$vector\_pair$ = $\{ \{S_0, 0\}, \{S_1, 1\}, \dots, \{S_{m-1}, m-1\} \}$
$S_{sum} = S_0 + S_1 + \cdots + S_{m-1}$
Temporary variables: $temp\_sum, bit\_count, i, j, k, check = 0$.
Step 1: Sort the $vector\_pair$ only based on the $S_i$ values.
Step 2:
for $i = 0$;  $i < m$ AND $check == 0$;  $i$++;
do
    $temp\_sum = 0$
    for $j = 0$;  $j <= i$;  $j$++;
    do
        $temp\_sum = temp\_sum + vector\_pair[\ j\ ].\ S_{value}$
        $bit\_count = j$
    end
    if $(temp\_sum > E_r \times S_{sum})$ then
        $check = 1$
        for $k = 0$;  $k <= bit\_count$;  $k$++;
        do
            Bit[ $vector\_pair[\ k\ ]$.index ] = 1;
        end
    end
end
```

Algorithm 2: Pseudocode to create a binary vector from a real valued vector in which there is no need to hard-code any conditions into the program.

We added more information about the relative largeness[4] of the encoded values into the binary descriptor with few more extra bits, but it did not improve the performance of B-SHOT, highlighting that the current binarization scheme holds the required information in a compact form.

3D feature descriptors are mainly employed to calculate approximate 3D transformation between a source and a target point cloud, which is used to initialize an Iterative Closest Point (Besl and McKay 1992) algorithm that performs fine and accurate registration. Hence, a marginal reduction in the number of correspondences is acceptable if the found cor-

[4] The way we added the extra information about the relative largeness and the experimental results are available at http://tinyurl.com/eb-shot.

respondences can roughly estimate the 3D transformation matrix.

4 Experimental evaluation

In this section, we apply the proposed binarization technique on three state-of-the-art 3D feature descriptors, SHOT, RoPS and FPFH and show that it offers good performance when applied on SHOT descriptor. Later, we provide an extensive performance evaluation of the B-SHOT binary feature descriptor in terms of its capability to estimate 3D transformation and quantitative evaluation of the established keypoint correspondences while comparing with the state-of-the-art real valued 3D feature descriptors.

The source code of B-SHOT and additional experimental results will be made available at https://sites.google.com/site/bshotdescriptor/.

4.1 Experimental setup

4.1.1 Computer specifications

In all our experiments, we have used a CPU with an *Intel Xeon(R) CPU E5-1650 0 @ 3.20GHz* × 12 and 16 GB RAM with *UBUNTU 14.04* operating system.

4.1.2 Dataset

We employ the publicly available Kinect dataset[5] (Tombari et al. 2013) for experimental evaluation and comparison of the proposed B-SHOT feature descriptor with the state-of-the-art real valued 3D feature descriptors. The Kinect dataset provides *models*, *scenes* and respective ground truth 3D transformations between them. In this dataset, *models* represent the objects whereas *scenes* contain a collection of various objects in different orientations and occlusions, with a back ground. The ground truth information provides the 3D transformation between a *scene* and an object *model* present in the considered *scene*. There are 17 scenes and 7 object models in this Kinect dataset. Each scene has about 3 models (objects) present in it and there are 49 scene-model samples in total (Fig. 3).

The relevance of the Kinect dataset to practical applications was also mentioned in Guo et al. (2015), as it has inherent noise that arises from the widely used Microsoft Kinect sensor. Another low cost hand-held 3D depth acquisition device from Google, Project Tango, also provides similar kind of noisy depth data as that of Microsoft's Kinect sensor. Hence, we perform extensive evaluation on the Kinect dataset and compare the proposed B-SHOT descriptor with SHOT, FPFH and RoPS feature descriptors and also with their binary counterparts. From the literature (Guo et al. 2015; Salti et al. 2014), SHOT turns out to be the state-of-the-art 3D feature descriptor while the FPFH feature descriptor and RoPS offer competitive performance (Guo et al. 2015). All these implementations are publicly available though the Point Cloud Library (Rusu et al. 2011).

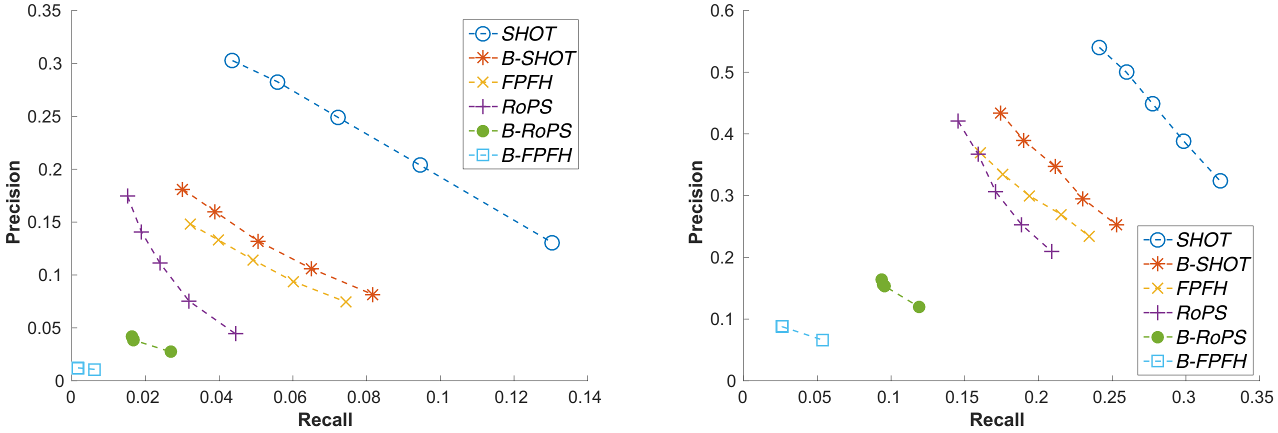
Fig. 3 A scene named `scene005` (*left*) and two models named `doll018` (*middle*) and `mario000` (*right*) from the Kinect dataset

4.2 On selecting B-SHOT over B-RoPS and B-FPFH

Guo et al. (2015) have presented an extensive evaluation of local 3D feature descriptors in various test scenarios and concluded that SHOT (Salti et al. 2014), RoPS (Guo et al. 2013) and FPFH (Rusu et al. 2009) feature descriptors offer good performance with some trade-off's in terms of descriptiveness, computational requirements and memory footprint. Hence, the proposed binarization technique was applied to SHOT, RoPS and FPFH feature descriptors to create their binary counterparts, namely, B-SHOT, B-RoPS and B-FPFH. Our experiments with various point clouds have shown that B-SHOT offers a better feature descriptor matching performance compared to B-RoPS and B-FPFH, i.e. B-SHOT finds more number of true keypoint correspondences when compared to B-RoPS and B-FPFH. To bolster this fact that B-SHOT offers better performance when compared to B-RoPS and B-FPFH, we employ conventional precision recall curve based evaluation (Salti et al. 2014; Guo et al. 2015) and also illustrate an interesting observation from the concept of distance distributions, as mentioned in BRIEF (Calonder et al. 2012), the first binary 2D feature descriptor. Moreover, we also show the number of true correspondences established by B-SHOT, B-RoPS and B-FPFH and show that B-SHOT offers more number of true correspondences.

Precision-recall curve based evaluation: Following the evaluation criterion used in state-of-the-art works on 3D feature descriptors, we employ precision-recall curves to compare SHOT, RoPS, FPFH, and their binary versions B-SHOT, B-RoPS and B-FPFH, on the Kinect dataset with 49 scene-model pairs. Firstly, we detect uniform keypoints on the model and then based on the available groundtruth 3D

[5] http://vision.deis.unibo.it/research/80-shot.



(a) Settings: Uniform keypoints are detected for every$0.01m$ and a support size of $0.12m$ was used to extract feature descriptors.

(b) Settings: Uniform keypoints are detected for every$0.02m$ and a support size of $0.08m$ was used to extract feature descriptors.

Fig. 4 Precision recall curves based evaluation of real valued feature descriptors, SHOT, RoPS, FPFH and their binary counterparts, B-SHOT, B-RoPS and B-FPFH. It can be observed from the above figure that the performance trend is as follows, SHOT > B-SHOT > FPFH > RoPS > B-RoPS > B-FPFH with the employed settings on the Kinect dataset

transformation, we find scene keypoints that correspond to the previously detected model keypoints. In this way, the 3D feature detector's inaccuracy in finding repeatable keypoints is unaccounted and only the ability of 3D feature descriptor's descriptiveness is evaluated. Then, 3D feature descriptors are calculated at the found model and scene keypoints, and the first and the second nearest scene feature descriptors for every model feature descriptor is found. If the ratio of the first to the second nearest neighbour is less than a threshold α, then it is considered as a match or else it is discarded.

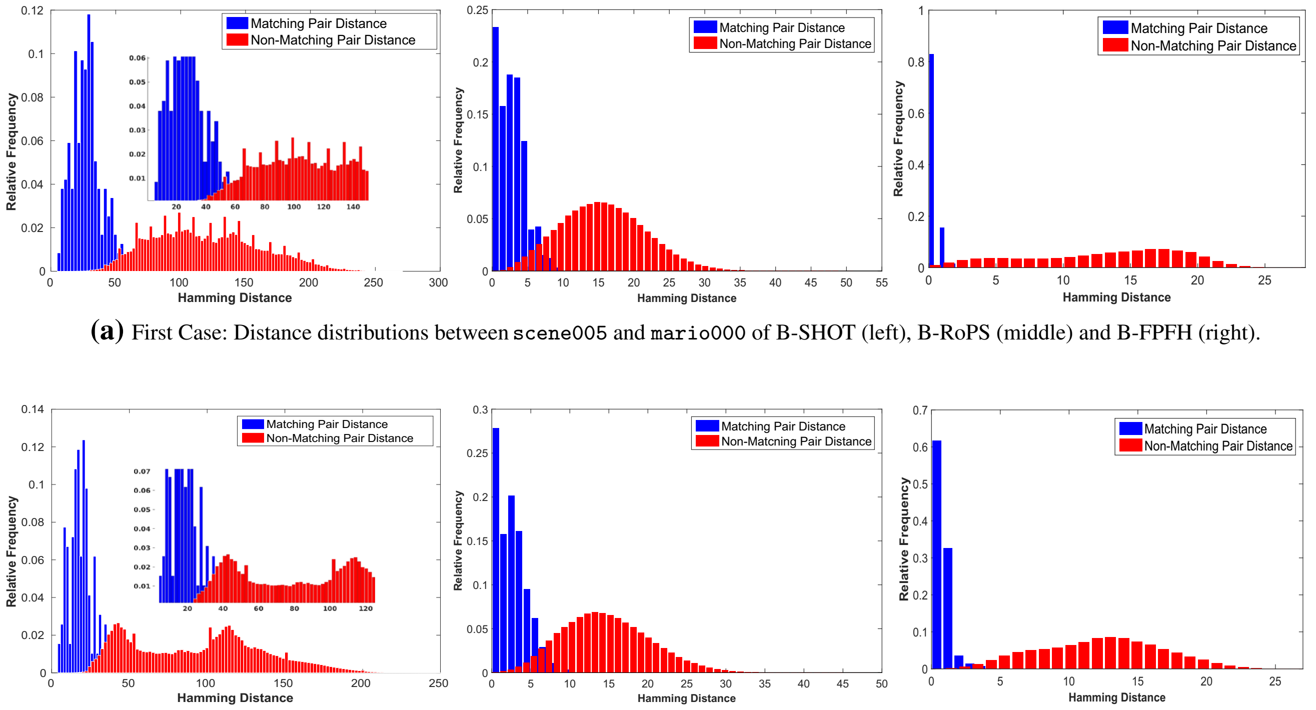
To calculate precision and recall, firstly the groundtruth matches G_M, which represent the number of model keypoints that lie on the scene are calculated. Secondly, the threshold α is varied and the feature descriptor matches, FD_α are found based on the above mentioned distance ratio test. Thirdly, the true matches, $True_\alpha$, are calculated by finding the number of FD_α that comply with the ground truth matches G_M. Finally, as mentioned by (Guo et al. 2015), precision and recall at each α is calculated as shown below.

$$Precision = \frac{True_\alpha}{FD_\alpha} \tag{2}$$

$$Recall = \frac{True_\alpha}{G_M} \tag{3}$$

In our experiments, we found that meaningful correspondences could be established for the threshold $\alpha > 0.9$. Hence, in our experiments, we used five values of $\alpha = \{0.9, 0.925, 0.95, 0.975, 1\}$ for evaluation. The precision and recall values at each value of α is estimated and averaged over 49 scene-model pairs. We performed two experiments by setting the uniform keypoint detection radius $R_{kp} = \{0.01\,\text{m}, 0.02\,\text{m}\}$ and the support size $R_{fd} = \{0.12\,\text{m}, 0.08\,\text{m}\}$ that is used for calculating feature descriptors. The results of these two experiments are shown in Fig. 4 and they highlight that B-SHOT offers second best performance when compared to all others, most importantly, B-SHOT is better slightly better than RoPS and FPFH. The performance trend from Fig. 4 can be summarized as SHOT > B-SHOT > FPFH > RoPS > B-RoPS > B-FPFH with the employed settings on the Kinect dataset.

Distance distributions: Here we perform the distance distributions experiment as mentioned in BRIEF (Calonder et al. 2012), a 2D binary descriptor. Accordingly, for this experiment, we consider the scene `scene005`, and the models `doll018` and `mario000` from the Kinect dataset, as shown in Fig 3. Let us consider `scene005` and `mario000` as the first case, and `scene005` and `doll018` as the second case. In both cases, we find keypoints with a uniform keypoint detector all over the scene and the model with 0.01 m radius. Then SHOT, FPFH and ROPS descriptors are extracted at all those keypoints with the support radius of 0.12 m and their corresponding B-SHOT, B-FPFH and B-RoPS binary feature descriptors are constructed. We match the constructed binary feature descriptors based on the Hamming distance metric and apply RANSAC (Fischler and Bolles 1981) to remove false correspondences. Finally, in Fig. 5, we show the distribution of Hamming distances between the matching keypoint pairs and the non-matching keypoint pairs in both the test cases with B-SHOT, B-FPFH and B-RoPS descriptors. The motivation behind performing the distance distribution experiments is that if the matching pair distance distribution and the non-matching pair distance distribution are well separated, the developed descriptors are highly discriminative (Calonder et al. 2012). Hence, a simple Hamming



(a) First Case: Distance distributions between `scene005` and `mario000` of B-SHOT (left), B-RoPS (middle) and B-FPFH (right).

(b) Second Case: Distance distributions between `scene005` and `doll018` of B-SHOT (left), B-RoPS (middle) and B-FPFH (right).

Fig. 5 Distance distributions of B-SHOT, B-RoPS and B-FPFH binary feature descriptors. As can be seen from **a** and **b**, the left most distance distributions in both cases, which represent B-SHOT, are well separated suggesting the high descriptiveness of B-SHOT when compared to B-RoPS and B-FPFH

distance based classifier can be employed to separate these two classes effectively.

It can be seen from Fig. 5 that in both the cases, as shown in Fig. 5a and b, B-SHOT's distance distributions are well separated whereas the distance distributions of B-RoPS and B-FPFH are not highly separable. It can be seen from Fig. 5a (left) that the overlap ends at a Hamming distance of 40 while in Fig. 5b (left) the overlap ends at a Hamming distance of 20. This can be seen by zooming the figures. Hence, based on this observation that there is no overlap for Hamming distances less than 20, a simple Hamming distance based classifier with a threshold of 20 can classify and produce only true correspondences in the case of B-SHOT. In the case of B-RoPS, the threshold for the Hamming distance metric has to be less than '2' to find true correspondences alone. For, B-FPFH, a simple classifier may not always produce true correspondences irrespective of the threshold in some of the cases, because there is overlap even at a Hamming distance of 0, as can be seen from Fig. 5a–b (right).

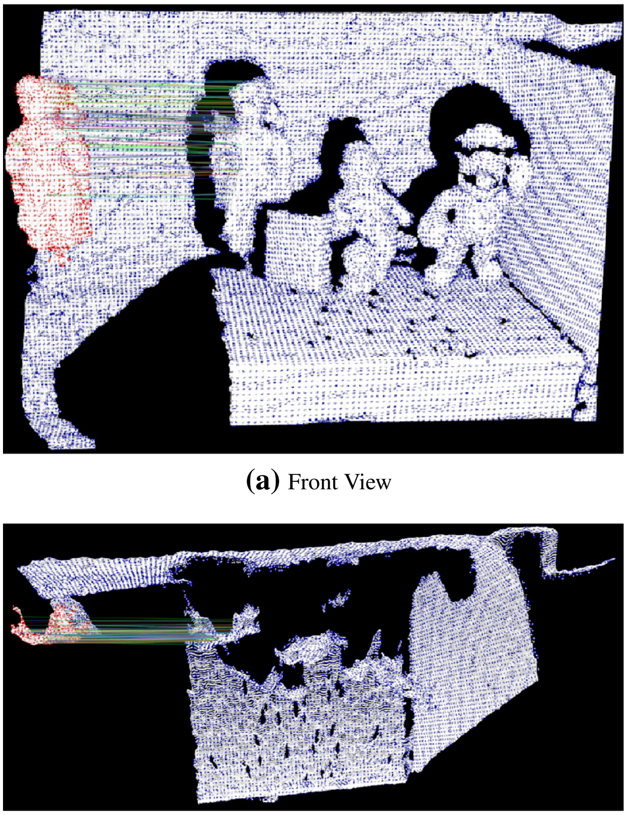
It should be noted that there are others factors forming the reasons behind B-SHOT offering good performance, when compared to B-RoPS and B-FPFH. Firstly, B-SHOT and B-RoPS are constructed from SHOT and RoPS, which are hybrid feature descriptors (Salti et al. 2014) that take the advantages of being a combination of signature and histogram based methods, whereas FPFH is just a histogram based method. Secondly, B-SHOT contains more information as it has 352 bits, while B-RoPS is 135 bits and B-FPFH is 33 bits in size, hence, making B-SHOT comparatively more descriptive.

Table 1 Number of true keypoint correspondences found between the scene `scene005`, and models `doll018`, and `mario000` using B-SHOT, B-RoPS and B-FPFH feature descriptors

scene005	*B-SHOT*	*B-RoPS*	*B-FPFH*
Keypoint Radius: 0.01 m			
Doll018 (909 keypoints)	**128**	74	28
Mario000 (860 keypoints)	**213**	135	43
Keypoint Radius: 0.02 m			
Doll018 (252 keypoints)	**57**	25	13
Mario000 (261 keypoints)	**94**	53	19
Keypoint Radius: 0.03 m			
Doll018 (123 keypoints)	**33**	17	8
Mario000 (125 keypoints)	**41**	21	15

A uniform keypoint detector with $R_{kp} = \{0.01\,\text{m}, 0.02\,\text{m}, 0.03\,\text{m}\}$ and feature descriptor support size of $R_{fd} = 0.10\,\text{m}$ was employed

Finally, in Table 1, we show the exact number of true keypoint correspondences established by B-SHOT, B-RoPS and B-FPFH binary feature descriptors. We consider two test cases as before, i.e., `scene005` and `doll018`,



(a) Front View

(b) Top View

Fig. 6 B-SHOT correspondences after RANSAC on a scene `scene005` and model `doll018` from the Kinect dataset as shown in Fig. 3. It can be seen that there is no false correspondence. Scene and model point clouds are represented in *white* colour while the keypoints detected on them in *blue* and *red* colours respectively (Color figure online)

and `scene005` and `mario000`. We employ uniform keypoint detector and vary the keypoint radius $R_{kp} = \{0.01\,\text{m}, 0.02\,\text{m}, 0.03\,\text{m}\}$ to detect a varied number of keypoints. As can be seen from Table 1, in both cases with a different number of keypoints, B-SHOT finds a higher number of true correspondences when compared to B-RoPS and B-FPFH. This also holds for other scene-model pairs in the Kinect dataset.

In Fig. 6, we illustrate the B-SHOT correspondences after the RANSAC algorithm between a scene and a model named `scene005` and `doll018`, from the Kinect dataset, which are shown in Fig. 3. In Fig. 6, uniform keypoints were detected with radius $R_{kp} = 0.01\,\text{m}$ and the employed B-SHOT descriptor support size was $R_{fd} = 0.12\,\text{m}$. As can be seen from the figure, there is no false correspondence after reciprocal nearest neighbour based feature matching and RANSAC. In the figure, the scene and model point clouds are represented in white colour while the keypoints detected on them are shown in blue and red colours respectively.

Based on these three experiments, the first one based on the precision recall curves, the second one based on distance distributions and the last one based on established true keypoint correspondences, it can be claimed that the proposed binarization technique works better when applied to SHOT rather than RoPS and FPFH.

4.3 A robust evaluation metric for 3D feature descriptor comparison

Existing works in the 2D image domain (Calonder et al. 2012; Yang and Cheng 2014) and 3D point cloud domain (Salti et al. 2014; Guo et al. 2015) employ *Recognition Rate* and *Precision vs Recall* curves for performance evaluation of feature descriptors. In both these experimental frameworks, they detect keypoints on the model, apply the ground truth 3D transformation and find the exact same keypoints on the scene. These keypoints are then matched via feature descriptors. In *Recognition Rate*, the number (or ratio) of correct feature matches are reported. *Precision vs Recall* curves are generated by varying the threshold that is used to establish the correspondences based on the ratio of the first and the second best nearest neighbours. In the existing evaluation frameworks (*Recognition Rate* and *Precision vs Recall*), there are two main drawbacks. Firstly, they do not consider the keypoint detection ambiguity which is significant in 3D domain.[6] Secondly, they apply the available groundtruth 3D transformation on the model keypoints to find the exactly same keypoints on the scene, hence they do not consider the false positives that may arise from the back ground of the scene in which models are present. Therefore we propose an evaluation metric that addresses the above-mentioned issues.

The two main important traits that are expected from a good 3D feature descriptor:

1. Ability to match the keypoints that arise from similar areas with small position ambiguity due to the imperfection of current 3D keypoint detectors.
2. Avoiding false positives that arise from back ground and from other models that are present in the scene.

Hence, we propose to employ a uniform keypoint detector to detect keypoints with the highest ambiguity in keypoint repeatability (to account for the worst case scenario). The uncertainty in the keypoint detection can be varied accordingly with the keypoint detection radius R_{kp}. Next, this uniform keypoint detector is used to find keypoints all over the scene and the model to allow for all the possible false correspondences from the back ground and other models

[6] State-of-the-art 3D keypoint detectors achieve at most 0.5 relative repeatability (Tombari et al. 2013), i.e., only half of the detected keypoints between a scene and a model lie exactly at the same positions.

present in the scene. Our experiments with SHOT, FPFH and RoPS have shown that even these real valued state-of-the-art 3D feature descriptors generate some false correspondences in almost all of the scenarios. It is therefore inevitable to use RANSAC (Fischler and Bolles 1981) based 3D transformation estimation, to remove those false correspondences. Moreover, the distance ratio based nearest neighbour matching, in general, is not employed in any real world application, instead the first nearest neighbour is assigned as a match and RANSAC is later employed to prune outliers.

Hence we propose a more robust evaluation metric to compare 3D feature descriptors as described below:

1. Detect uniform 3D keypoints with a voxel grid filter of radius R_{kp} on both the scene and the model independently. This caters for the possible ambiguity from 3D keypoint detection, which can be varied accordingly with R_{kp} and considers all the possible false correspondences from back ground of the scene as well.
2. Extract 3D feature descriptors with a support radius of R_{fd} around the detected keypoints and establish nearest neighbour correspondences that are reciprocal.
3. Apply RANSAC to find the final set of correspondences and estimate the 3D transformation T_R using the maximal consensus set.

We then compute the difference, T_{diff}, between the estimated 3D transformation via RANSAC, T_R, and the available ground truth transformation, T_{GT}, by employing the Euclidean metric as shown below:

$$T_{diff} = \sqrt{\sum_{i=0}^{n}\sum_{j=0}^{n}(T_{R_{ij}} - T_{GT_{ij}})^2} \tag{4}$$

where $n = 3$ in the case of 3D homogeneous transformation matrices, and $T_{R_{ij}}$ and $T_{GT_{ij}}$ represent the corresponding elements at indices $\{i, j\}$ in the transformation matrices.

The T_{diff} error metric provides a quantitative measure of how well the established correspondences fulfil the purpose of accurately estimating the 3D transformation between the *scene* and the *model*. A greater value or a peak in T_{diff} implies that the estimated 3D transformation is considerably different from the groundtruth highlighting the possibility of no true keypoint correspondences.

Once the 3D transformation is estimated, the next step is to evaluate how many detected keypoints are positively matched via feature descriptors. To quantify this, we find the number of true RANSAC correspondences by verifying their compliance with the available groundtruth transformation. We transform the detected scene keypoints onto the model by applying the groundtruth transformation and check if the established RANSAC keypoint correspondences lie within a neighbourhood of ϵ. If a RANSAC keypoint correspondence lies within this ϵ, it is considered to be a true RANSAC match. We provide the ratio of true RANSAC matches to the number of detected keypoints that are present in both the scene and the model, and term it as robust recognition rate (RRR). To find the number of keypoints present in both the scene and the model, we use the same method as employed in repeatability tests (Tombari et al. 2013), wherein, we transform the model keypoints onto the scene and verify if there is any keypoint present in the scene within a neighbourhood of ϵ. If yes, it is considered to be a keypoint that is present in both the scene and the model.

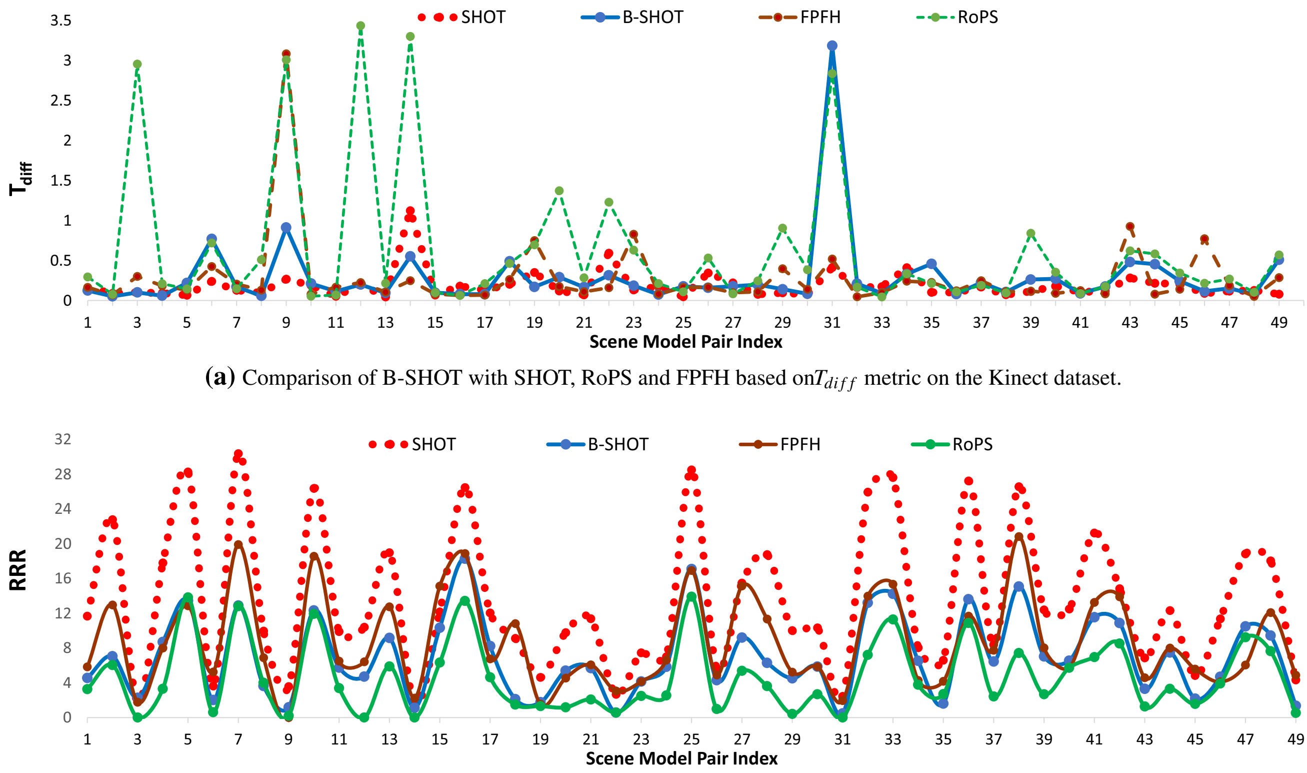
RRR can be seen as an enhancement to the *Recognition Rate* that was proposed in (Calonder et al. 2012, 2010). In short, there are three enhancements, firstly, in *RRR*, the ambiguity in 3D keypoint detection is considered by employing a uniform keypoint detector. Secondly, the feature descriptor's capability in avoiding all possible false correspondences is evaluated as the keypoints are detected on both the model and scene independently without applying the groundtruth transformation. Lastly, a more realistic setting is added where the first nearest neighbours are assigned as the matches and RANSAC is employed to prune outliers. Then, robust recognition rate (RRR) is estimated as shown below.

$$RRR = \frac{\textit{True matches in RANSAC correspondences} \times 100}{\textit{Number of keypoints in both model and scene}} \tag{5}$$

The T_{diff} and *RRR* metrics complement each other in providing substantial information about the feature descriptor's performance. A low value in the T_{diff} metric at a specific scene-model pair index highlights that the estimated transformation is close to the groundtruth while the corresponding *RRR* metric at the same scene-model pair index provides a quantitative measure of the established true correspondences. A peak in the T_{diff} metric highlights a potential failure case and the corresponding *RRR* metric goes to zero, as there will be no true correspondences. In general, we found that a T_{diff} value greater than 1.5 or 2 is a failure case where the feature descriptors cannot establish any true feature correspondences and correspondingly the *RRR* metric goes to zero. It is easy to infer a successful case by looking solely at the *RRR* metric, as it would have a high *RRR* value. But, when the *RRR* value is close to zero, then the T_{diff} metric turns out to be helpful in concluding if the estimated transformation is correct or not.

4.4 Comparison of B-SHOT with SHOT, RoPS and FPFH

In this section, we evaluate and compare B-SHOT with other real valued feature descriptors based on T_{diff} and *RRR* metrics. T_{diff} evaluates if the found correspondences successfully



(a) Comparison of B-SHOT with SHOT, RoPS and FPFH based onT_{diff} metric on the Kinect dataset.

(b) Comparison of B-SHOT with SHOT, RoPS and FPFH based on*RRR* metric on the Kinect dataset.

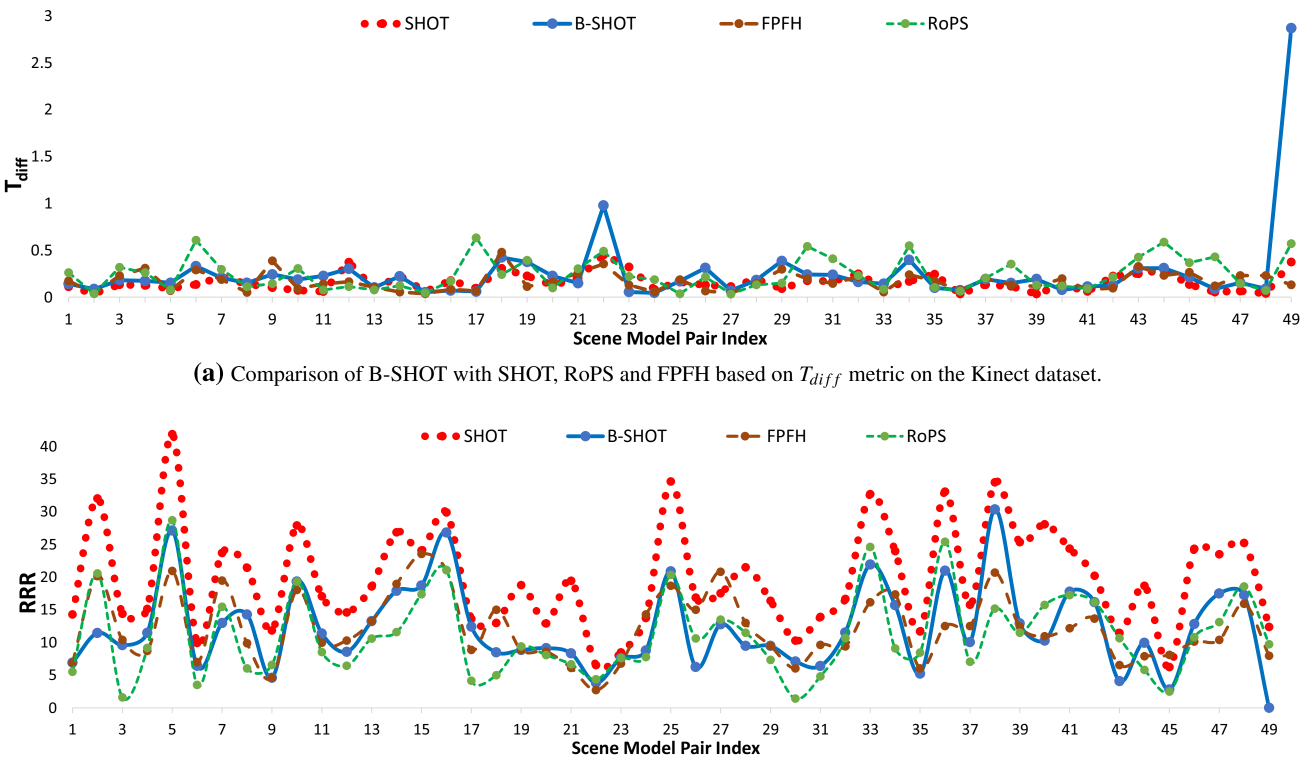
Fig. 7 Comparison of B-SHOT with SHOT, RoPS and FPFH based on the T_{diff} and *RRR* metrics. Parameters: $R_{kp} = 0.01$ m and $R_{fd} = 0.12$ m. These values of $R_{kp} = 0.01$ m and $R_{fd} = 0.12$ m highlight the possibility where the detected keypoints are very close and there is large overlap and interference between the neighbourhood keypoint descriptors. A peak in the T_{diff} metric highlights a failure case where there is a significant error between the estimated 3D transformation and the available groundtruth. A value close to zero in *RRR* metric highlights that the number of true RANSAC correspondences are much lesser when compared to the detected keypoints

estimate the 3D transformation between the scene and the model, whereas *RRR* provides the ratio of correspondences found between the scene and the model. Through these experiments, it can observed that there is a small drop in the number of correspondences found using B-SHOT when compared to SHOT, however, the found correspondences are good enough to estimate the 3D transformation, which is the main application of 3D feature descriptors.

We perform two experiments on the Kinect dataset with different parameter settings for keypoint detection and feature description. In experiment 1, the keypoints are close to each other and the neighbouring feature descriptors have large overlap, while in experiment 2, keypoints are farther away with higher position ambiguity when compared to feature descriptor's support size.

Experiment 1: In Fig. 7, we compare B-SHOT with the state-of-the-art real valued feature descriptors, SHOT, RoPS and FPFH, based on the T_{diff} (Eq. 4) and *RRR* (Eq. 5) metrics on the Kinect dataset. The keypoints were detected using a voxel grid filter (uniform keypoint detector) with leaf size, $R_{kp} = 0.01$ m and feature descriptors were extracted with support size, $R_{fd} = 0.12$ m. Reciprocal nearest neighbour correspondences between scene and model keypoints are established by feature descriptor matching. The Euclidean distance metric is employed for real valued feature descriptors, SHOT, FPFH and RoPS and the Hamming distance metric is used for the B-SHOT. Finally, the best consensual set of matches are found using the RANSAC algorithm and the T_{diff} and *RRR* metrics are estimated. We perform these steps on all 49 scene-model pairs in the Kinect dataset. In Fig. 7, the employed parameter values of $R_{kp} = 0.01$ m for uniform keypoint detector, and $R_{fd} = 0.12$ m for descriptor support radius, highlight the scenario where the detected keypoints are close to each other, have less position ambiguity, while having large overlap and interference between the neighbouring keypoint descriptors.

It can be seen from the T_{diff} metric, as shown in Fig. 7a, that RoPS fails in more cases when compared to others. It can be simultaneously observed from the *RRR* metric, (Fig. 7b),



(a) Comparison of B-SHOT with SHOT, RoPS and FPFH based on T_{diff} metric on the Kinect dataset.

(b) Comparison of B-SHOT with SHOT, RoPS and FPFH based on *RRR* metric on the Kinect dataset.

Fig. 8 Comparison of B-SHOT with SHOT, RoPS and FPFH based on the T_{diff} and *RRR* metrics. Parameters: R_{kp} = 0.02 m and R_{fd} = 0.08 m. These values of R_{kp} = 0.02 m and R_{fd} = 0.08 m highlight the possibility where the detected keypoints are bit further when compared to the feature descriptor's support size. It can be seen that B-SHOT fails only in one case (index value = 49) while performing well on all other scene-model pairs. **a** Comparison of B-SHOT with SHOT, RoPS and FPFH based on T_{diff} metric on the Kinect dataset. **b** Comparison of B-SHOT with SHOT, RoPS and FPFH based on *RRR* metric on the Kinect dataset

that at those corresponding indices where T_{diff} has peaks (for example, index = {3, 12}), the *RRR* value of RoPS goes to zero, indicating that there are no true correspondences. B-SHOT fails only in one case (index = 31) out of the 49 scene-model pairs. Though B-SHOT has slightly higher T_{diff} values at indices 6 and 9, they are not failure cases as the corresponding *RRR* values are not exactly zero, highlighting the presence of few true correspondences. From Fig. 7b, it can be seen that RoPS offers lower true RANSAC correspondences ratio when compared to B-SHOT, SHOT and FPFH descriptors on the Kinect dataset with these parameter settings. B-SHOT offers a lower ratio of true correspondences when compared to SHOT and FPFH because of the previously mentioned characteristics of being a binary 3D feature descriptor. If *RRR* metric alone is seen (Fig. 7b), it may give an impression that B-SHOT offers a lower number of true RANSAC correspondences ratio, but through the T_{diff} metric (Fig. 7a) it can be ensured that this decrease in feature correspondences does not affect the main purpose of 3D transformation estimation via feature descriptor matching. Once a rough 3D transformation is estimated, ICP algorithm (Besl and McKay 1992) can be used to perform fine and accurate registration, which inherently provides more number of correspondences if required.

Experiment 2: In this experiment, keypoints were detected with R_{kp} = 0.02 m and feature descriptors were extracted with support size, R_{fd} = 0.08 m. This resembles a scenario where the detected keypoints (R_{kp} = 0.02 m) are farther when compared to the employed descriptor's support size (R_{fd} = 0.08 m), while having greater position ambiguity in keypoint detection and lesser overlap in neighbouring feature descriptors. We perform the same steps of finding true RANSAC matches and the consensual transformation to estimate T_{diff} and *RRR* metrics for all the scene-model pairs in the Kinect dataset.

It can be seen from the T_{diff} metric, as shown in Fig. 8a, that B-SHOT fails in one case (index = 49) while working well in all other cases. Figure 8b shows that B-SHOT's *RRR* value goes to zero at index = 49. The reason behind this is that the created B-SHOT descriptors were not discriminative enough to find true correspondences in this specific scene-model pair. Except the one failure case, B-SHOT's performance based on the *RRR* metric is comparable to RoPS in this scenario. In comparison with the previous settings, it can be seen that

Table 2 Robust recognition rate (RRR) as offered by various feature descriptors in experiments 1 & 2 on the Kinect dataset

Descriptor	$RRR_{Exp\ 1}$	$RRR_{Exp\ 2}$
SHOT	13.43	19.78
FPFH	8.6	12.20
B-SHOT	7.03	12.20
RoPS	4.59	11.16

the *RRR* value of B-SHOT and other feature descriptors is higher because there is less interference in the neighbourhood keypoint descriptors.

FPFH offered good feature matching performance on the Kinect dataset but its computational time increases exponentially with the increase in the support radius,[7] Guo et al. (2015) through their extensive performance evaluation, highlighted that FPFH is preferable when the points in the local neighbourhood are relatively less. Therefore FPFH is not preferable in terms of computational requirements when the point clouds are dense and the support size for the feature descriptor is relatively bigger. It turns out that in our experiments, B-SHOT offers better performance than the RoPS feature descriptor on the Kinect dataset with the employed settings.

In Table 2, we present the averaged *RRR* values offered by SHOT, FPFH, B-SHOT and RoPS descriptors on the Kinect dataset in experiments 1 & 2. Please note that in both these experiments the normal estimation radius is set to 0.02 m while the $RANSAC_{inlier\ threhsold}$ is set to 0.01. It can be seen from Table 2 that SHOT established more number of true correspondences, followed by FPFH, B-SHOT and the RoPS.

4.5 Performance evaluation in 3D object recognition scenario

In Fig. 9, we present a generic 3D object recognition pipeline (Aldoma et al. 2012, 2015) in which, 3D keypoints are first detected on both the object and the scene. Then 3D feature descriptors are extracted at these keypoints with a fixed support size and nearest neighbours are assigned as possible keypoint correspondences between the object and the scene. In the next stage to remove false positive feature matches, either RANSAC algorithm or a global hypothesis verification[8] (Aldoma et al. 2015) can be used. In these set of experiments, we employed a uniform keypoint detector for 3D keypoint detection and RANSAC algorithm for the removal of false positive correspondences. If the resultant matches comply with the available groundtruth from the dataset, then the object is successfully recognized and later if needed, its accurate pose can also be estimated using ICP algorithm (Besl and McKay 1992). It can be observed that this generic 3D object recognition pipeline with RANSAC is very similar to the robust evaluation metric introduced before. Specifically, a wrongly recognized object would result in high T_{diff} value. To evaluate the recognition performance of various 3D feature descriptors, we report the number of correctly detected object models and the total number of scene-model (object) pairs evaluated in the dataset.

For this 3D object recognition evaluation, we employ three datasets, namely Kinect, Random Views and Retrieval datasets (Tombari et al. 2013; Guo et al. 2015). The evaluation of SHOT, RoPS, FPFH and B-SHOT descriptors is firstly performed on the Kinect dataset with the same parameters for keypoint detection and feature description as in Experiments 1 & 2. The total number of scene-model pair evaluations performed are 49 in each of these experiments and the pairs in which the object is successfully recognized is reported in Table 3. We next experimented with the Random Views dataset, which is a synthetic dataset and has an artificially added Gaussian noise of 0.1 mesh resolution in the scenes. Uniform keypoints were found with $R_{kp} = 0.005$ m and feature descriptors were extracted with $R_{fd} = 0.10$ m. As previously mentioned, RANSAC is applied on the reciprocal nearest neighbours and the number of correctly recognized input object models among the 108 scene-model pairs is presented in Table 3. Finally, we also employed the Retrieval dataset and used the scenes with a relatively higher Gaussian noise of 0.5 mesh resolution. In this dataset, the scenes and the models are exactly same, except that the scenes are corrupted with noise and are in different orientations. The keypoint detection radius R_{kp} was set to 0.01 m while the descriptor support radius R_{fd} was set to 0.10 m.

Table 3 illustrates the number of scene-model pair evaluations performed in each dataset and the number of correctly recognized input object models. It can be seen from Table 3 that B-SHOT offers almost similar recognition performance as the SHOT descriptor on all the three datasets. On the Kinect dataset which holds more relevance to practical scenarios, B-SHOT successfully recognized all input object models except in one case. On the synthetic Random Views and Retrieval datasets, B-SHOT still offered very close recognition performance to the SHOT descriptor while RoPS offered best recognition performance. From this set of recognition experiments, it can be seen that B-SHOT offers competitive recognition performance and its relatively less loss in descriptivity does not significantly affect the recognition performance.

[7] This can also be seen from Fig. 9 of Salti et al. (2014).

[8] 3D Object Recognition based on Correspondence Grouping http://pointclouds.org/documentation/tutorials/correspondence_grouping.php.

Table 3 This table presents the number of evaluated scene-model pairs and the number of input object models which were correctly recognized in three datasets by the considered 3D feature descriptors

Dataset	No. of Scene-Models	*B-SHOT*	*SHOT*	*RoPS*	*FPFH*
Kinect: Exp1	49	48	49	44	48
Kinect: Exp2	49	48	49	49	49
Random Views	108	39	42	47	15
Retrieval	18	13	13	15	11

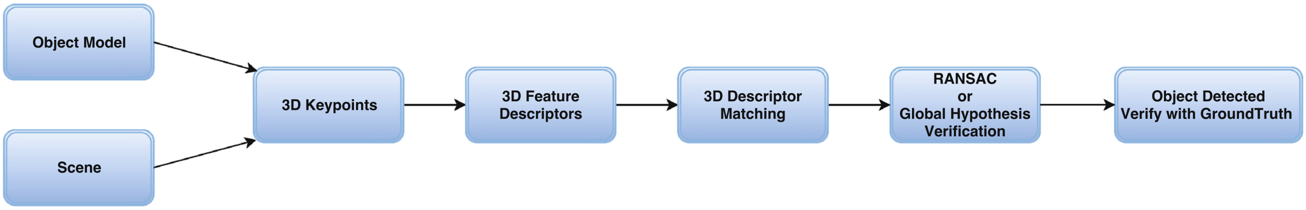
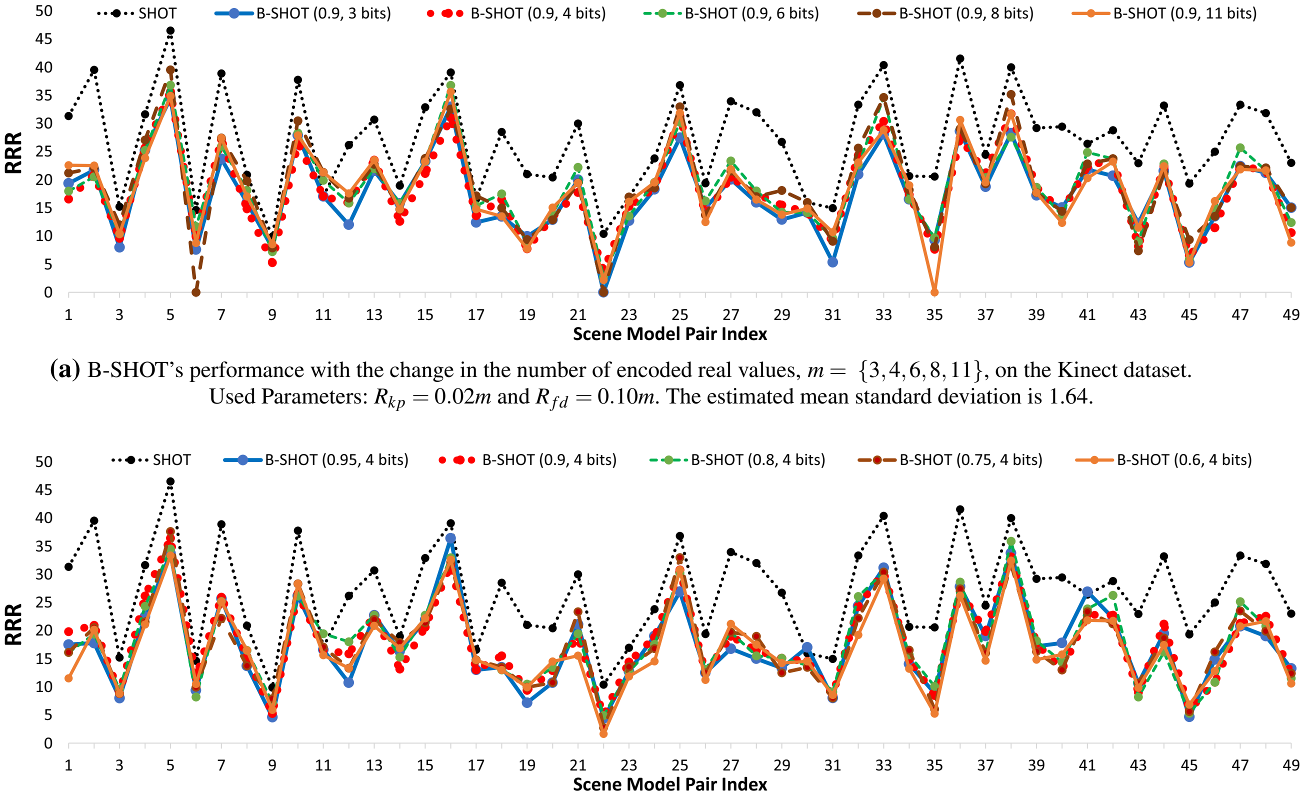

Object Model
Scene
3D Keypoints
3D Feature Descriptors
3D Descriptor Matching
RANSAC or Global Hypothesis Verification
Object Detected Verify with GroundTruth

Fig. 9 A generic 3D object recognition pipeline



(a) B-SHOT's performance with the change in the number of encoded real values, $m = \{3, 4, 6, 8, 11\}$, on the Kinect dataset. Used Parameters: $R_{kp} = 0.02m$ and $R_{fd} = 0.10m$. The estimated mean standard deviation is 1.64.

(b) B-SHOT's performance with the change in encoding ratio, $E_r = \{0.95, 0.9, 0.8, 0.75, 0.6\}$, on the Kinect dataset. Used Parameters: $R_{kp} = 0.02m$ and $R_{fd} = 0.10m$. The estimated mean standard deviation is 1.53.

Fig. 10 Performance of B-SHOT with the change in free parameters, m (number of encoded real values) and E_r (encoding ratio) evaluated using *RRR* metric on the Kinect dataset. **a** B-SHOT's performance with the change in the number of encoded real values, $m = \{3, 4, 6, 8, 11\}$, on the Kinect dataset

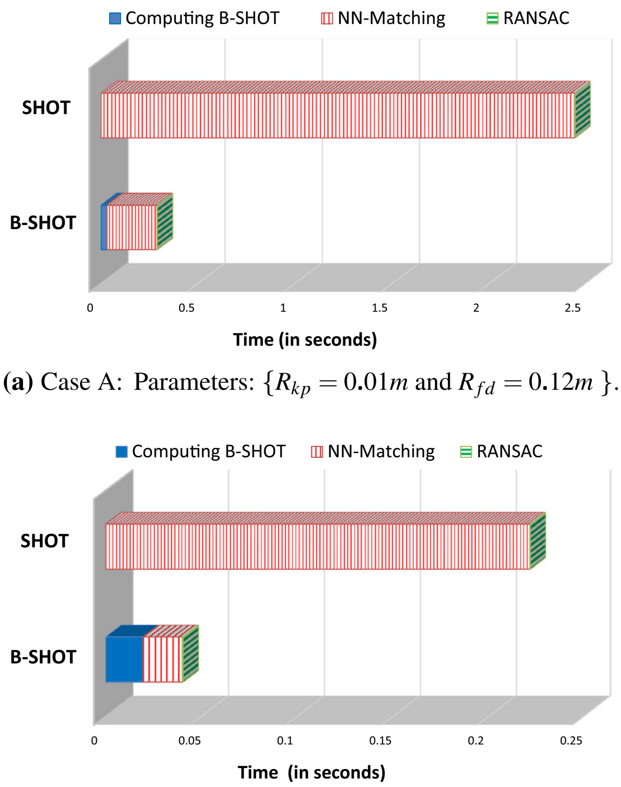
5 Parameter evaluation

The proposed binarization technique in Sect. 3 that is used to convert a real valued feature descriptor to a binary feature descriptor has two free parameters. Firstly, m, the number of real values that are encoded into binary values. Secondly, the encoding ratio, E_r, which is employed to quantize the real values to create binary representation. In all the experiments performed in the previous section, m was set to 4 and the encoding ratio, E_r was set to 0.9. In this section, we vary these two parameters, m and E_r, and show the performance of B-SHOT. We use the *RRR* metric, which represents the ratio of true RANSAC matches to the actual keypoints to evaluate the performance of B-SHOT with the change in the parameters, m and E_r. For these parameter evaluation experiments, we use the Kinect dataset, extract keypoints with $R_{kp} = 0.02$ m and feature descriptors with $R_{fd} = 0.10$ m while the false correspondences are removed with RANSAC.

In Fig. 10a, we show the performance of B-SHOT while varying the parameter $m = \{3, 4, 6, 8, 11\}$ and keeping $E_r = 0.9$ as a constant. For reference, we also show the performance of SHOT feature descriptor with the same settings of $R_{kp} = 0.02$ m and $R_{fd} = 0.10$ m in the figure. It can be seen from Fig. 10a that there is no significant variation but a minimal change (SD = 1.64) in B-SHOT's performance with the change in the parameter m. In particular, there is a drop in the performance, in the cases where $m = 8$ and $m = 11$ at index values of 6 and 35, correspondingly. There is a slight drop in the performance at index values of 12 and 31, when $m = 3$. In general, $m = \{4, 6\}$ offered similar performance and can be considered as a good choice for the proposed binarization technique to create B-SHOT from SHOT.

In Fig. 10b, we show the performance of B-SHOT while varying the values of $E_r = \{0.95, 0.9, 0.8, 0.75, 0.6\}$ and keeping $m = 4$ as a constant. The values of R_{kp} and R_{fd} are exactly the same as above where we change the parameter m. It can be seen from Fig. 10b that there is no significant variation (SD = 1.53) in the performance of B-SHOT with the change in the E_r parameter, except for small fluctuations when $E_r = 0.95$ and $E_r = 0.6$ at the index values of 19 and 22, correspondingly.

From this extensive performance evaluation based on T_{diff} and *RRR* metrics, evaluating the performance of B-SHOT with the change in free parameters, m and E_r and considering both the success and failure scenarios, it can be claimed that B-SHOT, being a binary descriptor, offers a competitive keypoint matching performance to the state-of-the-art real valued 3D feature descriptors. However, B-SHOT's advantages of having a very low memory footprint with faster descriptor matching capabilities outweigh its loss in descriptiveness.



(a) Case A: Parameters: $\{R_{kp} = 0.01m \text{ and } R_{fd} = 0.12m\}$.

(b) Case B: Parameters: $\{R_{kp} = 0.02m \text{ and } R_{fd} = 0.08m\}$.

Fig. 11 Computational time (in seconds) for B-SHOT and SHOT feature descriptor based matching of keypoints with different parameters on the Kinect dataset

Table 4 Computational time (in seconds) required for matching B-SHOT and SHOT feature descriptors

	Computing B-SHOT	*NN-Matching*	*RANSAC*
Case A			
B-SHOT	0.03117	0.25651	0.00052
SHOT	$(N.A)$	2.44355	0.00028
Case B			
B-SHOT	0.01943	0.02051	0.00036
SHOT	$(N.A)$	0.22121	0.0002

6 Computational and memory requirements

One of the biggest advantages of a binary feature descriptor is that feature descriptor matching can be performed extremely quickly. B-SHOT feature descriptors are matched using the hamming distance metric. The hamming distance between two binary vectors can be computed as the bitcount of the vector that results from the XOR operation of those two input binary vectors. This can be computed even faster on CPU's that have SSE 4.1 enabled, which supports the POPCNT instruction.

We compare the computational time required for SHOT and B-SHOT descriptor based matching of keypoints in Fig. 11 and Table 4. As the B-SHOT descriptor is created from the SHOT descriptor, the extra time required to create B-SHOT from SHOT is also considered in these experiments as shown in Fig. 11. Similar to previous experiments as performed in Figs. 7 and 8, two cases with exactly same parameter settings on the Kinect dataset are considered for computational time evaluation. In *Case A*, as shown in Fig. 11a, R_{kp} was set to 0.01 m and R_{fd} was set to 0.12 m. In the *Case B*, R_{kp} = 0.02 m and R_{fd} = 0.08 m and the computational time is shown in Fig. 11b. The number of detected keypoints is much lower in *Case B* when compared to *Case A* because of the value of R_{kp}. The time taken for computing B-SHOT from SHOT, matching B-SHOT descriptors between the model and the scene keypoints, and finding the final correspondences via RANSAC are all computed and averaged over the 49 samples in the Kinect dataset. Please note that the nearest neighbours for SHOT feature descriptor matching were obtained by creating a kdtree representation[9] and retrieving only the reciprocal correspondences.

It can be observed from Fig. 11 that the time taken for B-SHOT computation and matching is much lesser than the SHOT feature descriptor based NN-matching alone. Table 4 shows the exact values of the computational times. *Case A* refers to that in Fig. 11a while *Case B* refers to that in Fig. 11b. In *Case A*, B-SHOT feature descriptor computation and matching with RANSAC is about 8.4 times faster than SHOT descriptor matching, whereas in *Case B*, B-SHOT is 5.2 times faster than SHOT descriptor matching. This is because, in *Case A*, there are a greater number of keypoints and hence SHOT descriptor matching was even more computationally intensive when compared to *Case B*. From these experiments, we can conclude that B-SHOT feature descriptor matching is around 6 times faster than SHOT descriptor matching, thereby having immense potential in applications such as online 3D object recognition (Guo et al. 2014a; Faulhammer et al. 2015) and large scale 3D point cloud retrieval applications (Choi et al. 2016).

In general there are two steps for 3D keypoint matching via feature descriptors, the first being feature descriptor extraction while the second is feature descriptor matching. B-SHOT offers significant boost in feature descriptor matching stage as it can be matched using Hamming distance metric. An important point to note is that, as both B-SHOT and SHOT feature descriptors have the same offset computation time for extracting SHOT feature descriptors, we did not show them in the figure and the table as it would overshadow the enhancement offered by B-SHOT in feature descriptor matching stage. The actual computation time required for extracting SHOT feature descriptors on a single scene model pair in *Case A* and *Case B* is 50.1 and 16.42 s respectively, while running on a single thread.

The SHOT implementation as available in Point Cloud Library (Rusu et al. 2011) comes with a multi-threaded implementation, and on the employed CPU for experimentation which had 12 CPU cores, the SHOT feature descriptor extraction required 10.4 and 3.5 s of wall clock time for *Case A* and *Case B* respectively. This SHOT feature descriptor extraction time also varies with the number of keypoints, i.e., in *Case A*, keypoints are extracted for every 0.01 m while in *CaseB*, they were extracted at 0.02 m and hence is the difference in extraction time. Moreover, the SHOT feature descriptor ported on GPU (Palossi et al. 2013; Hu and Nooshabadi 2015), can offer 40×–54× speed up in feature descriptor extraction and hence feature descriptor extraction is not a bottleneck with available multi-core CPUs and GPUs. More importantly, in the applications such as large scale retrieval or 3D mobile visual search where feature descriptors are extracted only once but feature descriptor matching is carried out multiple times with a database of point clouds, faster descriptor matching and low memory footprint as offered by B-SHOT turn out to be more advantageous.

As mentioned earlier, binary descriptors have an edge over conventional feature descriptors in the memory required to store and represent them. While SHOT requires 1408 bytes (as a float value requires 4 bytes according to IEEE 754 single-precision binary floating-point format), as it is a 352 dimensional vector with each value ranging from 0 to 1, B-SHOT requires only 352 bits of binary data for its representation. There is a 32-fold reduction in the memory required to represent and store B-SHOT feature descriptors when compared to SHOT feature descriptors. This can be helpful in applications that require online transfer of feature descriptors as binary feature descriptors require much lesser bandwidth.

7 Conclusion

In this paper, we have introduced the very first binary 3D feature descriptor, B-SHOT, for fast keypoint matching on 3D point clouds. Specifically, a binarization method is proposed to convert a real valued feature descriptor to a binary feature descriptor. We applied the proposed binarization onto three state-of-the-art real valued 3D feature descriptors and show qualitatively and quantitatively that it offers best performance when applied to the SHOT (Salti et al. 2014) feature descriptor. Later, we introduced a robust metric for 3D feature descriptor evaluation, called, Robust Recognition Rate, by considering all possible false feature correspondences and catering for the ambiguity in 3D keypoint detection. Our experiments showed that the proposed B-SHOT offers comparable keypoint matching performance while having 32-fold less memory footprint and is approximately 6 times faster in feature descriptor matching. This work highlights that binary 3D feature descriptors are feasible and opens up a research direction for creating highly efficient binary 3D feature descriptors.

[9] We employ pcl::registration::CorrespondenceEstimation class from Point Cloud Library (www.pointclouds.org) to estimate reciprocal correspondences, which inherently uses a kdtree for faster matching and retrieval.

References

Alahi, A., Ortiz, R., & Vandergheynst, P. (2012). FREAK: Fast retina keypoint. In *2012 IEEE conference on computer vision and pattern recognition (CVPR)*.

Albarelli, A., Rodola, E., & Torsello, A. (2010). A game-theoretic approach to fine surface registration without initial motion estimation. In *2010 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 430–437). IEEE.

Albarelli, A., Rodola, E., & Torsello, A. (2010). Loosely distinctive features for robust surface alignment. In *Computer vision—ECCV 2010* (pp. 519–532). Springer.

Aldoma, A., Marton, Z. C., Tombari, F., Wohlkinger, W., Potthast, C., Zeisl, B., et al. (2012). Point cloud library: Three-dimensional object recognition and 6 DoF Pose Estimation. *IEEE Robotics & Automation Magazine*, *1070*(9932/12).

Aldoma, A., Tombari, F., Di Stefano, L., & Vincze, M. (2015). A global hypothesis verification framework for 3D object recognition in clutter. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PP*(99), 1–1. doi:10.1109/TPAMI.2015.2491940.

Besl, P. J., & McKay, H. D. (1992). A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *14*(2), 239–256.

Calonder, M., Lepetit, V., Ozuysal, M., Trzcinski, T., Strecha, C., & Fua, P. (2012). BRIEF: Computing a local binary descriptor very fast. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(7), 1281–1298. doi:10.1109/TPAMI.2011.222.

Calonder, M., Lepetit, V., Strecha, C., & Fua, P. (2010). Brief: Binary robust independent elementary features. In *Computer vision—ECCV 2010* (pp. 778–792). Springer.

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

Choi, S., Zhou, Q. Y., & Koltun, V. (2015). Robust reconstruction of indoor scenes. In *IEEE conference on computer vision and pattern recognition (CVPR)*.

Choi, S., Zhou, Q. Y., Miller, S., & Koltun, V. (2016). A large dataset of object scans. arXiv:1602.02481

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

Darom, T., & Keller, Y. (2012). Scale-invariant features for 3-D mesh models. *IEEE Transactions on Image Processing*, *21*(5), 2758–2769. doi:10.1109/TIP.2012.2183142.

Endres, F., Hess, J., Engelhard, N., Sturm, J., Cremers, D., & Burgard, W. (2012). An evaluation of the RGB-D SLAM system. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)* (pp. 1691–1696).

Faulhammer, T., Aldoma, A., Zillich, M., & Vincze, M. (2015). Temporal integration of feature correspondences for enhanced recognition in cluttered and dynamic environments. In *2015 IEEE International Conference on robotics and automation (ICRA)* (pp. 3003–3009). doi:10.1109/ICRA.2015.7139611

Fiolka, T., Stückler, J., Klein, D. A., Schulz, D., & Behnke, S. (2012). SURE: Surface entropy for distinctive 3D features. In *Spatial cognition VIII* (pp. 74–93). Springer.

Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, *24*(6), 381–395. doi:10.1145/358669.358692.

Frome, A., Huber, D., Kolluri, R., Bülow, T., & Malik, J. (2004). Recognizing objects in range data using regional point descriptors. In *Computer vision-ECCV 2004* (pp. 224–237). Springer.

Galvez-Lopez, D., & Tardos, J. (2012). Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, *28*(5), 1188–1197. doi:10.1109/TRO.2012.2197158.

Gong, Y., Lazebnik, S., Gordo, A., & Perronnin, F. (2013). Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(12), 2916–2929.

Guo, Y., Bennamoun, M., Sohel, F., Lu, M., & Wan, J. (2014). 3D object recognition in cluttered scenes with local surface features: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *36*(11), 2270–2287. doi:10.1109/TPAMI.2014.2316828.

Guo, Y., Bennamoun, M., Sohel, F., Lu, M., Wan, J., & Kwok, N. (2015). A comprehensive performance evaluation of 3D local feature descriptors. *International Journal of Computer Vision* (pp. 1–24). doi:10.1007/s11263-015-0824-y.

Guo, Y., Sohel, F., Bennamoun, M., Lu, M., & Wan, J. (2013). Rotational projection statistics for 3D local surface description and object recognition. *International Journal of Computer Vision*, *105*(1), 63–86. doi:10.1007/s11263-013-0627-y.

Guo, Y., Sohel, F., Bennamoun, M., Wan, J., & Lu, M. (2014). An accurate and robust range image registration algorithm for 3D object modeling. *IEEE Transactions on Multimedia*, *16*(5), 1377–1390. doi:10.1109/TMM.2014.2316145.

Hu, L., & Nooshabadi, S. (2015). G-SHOT: GPU accelerated 3D local descriptor for surface matching. *Journal of Visual Communication and Image Representation*, *30*, 343–349.

Johnson, A. E., & 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.

Knopp, J., Prasad, M., Willems, G., Timofte, R., & Van Gool, L. (2010). Hough transform and 3D SURF for robust three dimensional classification. In *Computer vision–ECCV 2010* (pp. 589–602). Springer.

Leutenegger, S., Chli, M., & Siegwart, R. Y. (2011). BRISK: Binary robust invariant scalable keypoints. In *2011 IEEE international conference on computer vision (ICCV)* (pp. 2548–2555). IEEE.

Leutenegger, S., Lynen, S., Bosse, M., Siegwart, R., & Furgale, P. (2015). Keyframe-based visual-inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, *34*(3), 314–334.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, *60*(2), 91–110.

Malaguti, F., Tombari, F., Salti, S., Pau, D., & Di Stefano, L. (2012). Toward compressed 3D descriptors. In *2012 second international conference on 3D imaging, modeling, processing, visualization and transmission (3DIMPVT)* (pp. 176–183). doi:10.1109/3DIMPVT.2012.9.

Marton, Z. C., Pangercic, D., Blodow, N., Kleinehellefort, J., & Beetz, M. (2010). General 3D modelling of novel objects from a single view. In *2010 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 3700–3705). IEEE.

Mian, A., Bennamoun, M., & 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*(2–3), 348–361.

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

Newcombe, R. A., Davison, A. J., Izadi, S., Kohli, P., Hilliges, O., Shotton, J., Molyneaux, D., Hodges, S., Kim, D., & Fitzgibbon, A. (2011). KinectFusion: Real-time dense surface mapping and tracking. In *IEEE International Symposium on Mixed and augmented reality (ISMAR)* (pp. 127–136).

Novatnack, J., & Nishino, K. (2008). Scale-dependent/invariant local 3D shape descriptors for fully automatic registration of multiple sets of range images. In *Computer vision—ECCV 2008* (pp. 440–453). Springer.

Palossi, D., Tombari, F., Salti, S., Ruggiero, M., Stefano, L., & Benini, L. (2013). GPU-SHOT: Parallel optimization for real-time 3D local description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 584–591).

Prakhya, S. M., Liu, B., & Lin, W. (2015). B-SHOT: A binary feature descriptor for fast and efficient keypoint matching on 3D point clouds. In *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*.

Prakhya, S. M., Liu, B., & Lin, W. (2016). Detecting keypoint sets on 3d point clouds via histogram of normal orientations. *Pattern Recognition Letters*. doi:10.1016/j.patrec.2016.06.002.

Prakhya, S. M., Liu, B., Lin, W., & Qayyum, U. (2015). Sparse depth odometry: 3D keypoint based pose estimation from dense depth data. In *2015 IEEE international conference on robotics and automation (ICRA)*

Project Tango. https://www.google.com/atap/project-tango/

Rodolà, E., Albarelli, A., Bergamasco, F., & Torsello, A. (2012). A scale independent selection process for 3d object recognition in cluttered scenes. *International Journal of Computer Vision*, *102*(1), 129–145.

Rodolà, E., Albarelli, A., Cremers, D., & Torsello, A. (2015). A simple and effective relevance-based point sampling for 3D shapes. *Pattern Recognition Letters*, *59*, 41–47.

Rusu, R., & Cousins, S. (2011). 3D is here: Point Cloud Library (PCL). In *2011 IEEE international conference on robotics and automation (ICRA)* (pp. 1–4). doi:10.1109/ICRA.2011.5980567

Rusu, R. B., Blodow, N., & Beetz, M. (2009). Fast point feature histograms (FPFH) for 3D registration. In *ICRA'09. IEEE international conference on robotics and automation, 2009* (pp. 3212–3217). IEEE.

Rusu, R. B., Blodow, N., Marton, Z. C., & Beetz, M. (2008). Aligning point cloud views using persistent feature histograms. In *IROS 2008. IEEE/RSJ international conference on intelligent robots and systems, 2008* (pp. 3384–3391). IEEE.

Salti, S., Tombari, F., & Di Stefano, L. (2014). SHOT: Unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding*, *125*, 251–264.

Steder, B., Rusu, R., Konolige, K., & Burgard, W. (2011). Point feature extraction on 3D range scans taking into account object boundaries. In *2011 IEEE international conference on robotics and automation (ICRA)* (pp. 2601–2608). doi:10.1109/ICRA.2011.5980187.

Strecha, C., Bronstein, A. M., M. M. B., & Fua, P. (2012). LDAHash: Improved matching with smaller descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(1).

Structure Sensor. http://structure.io/

Sturm, J., Bylow, E., Kahl, F., & Cremers, D. (2013). CopyMe3D: Scanning and printing persons in 3D. In *Pattern recognition, Lecture Notes in Computer Science* (pp. 405–414).

Tra, A. T., Lin, W., & Kot, A. (2015). Dominant SIFT : A novel compact descriptor. In *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)*.

Tabia, H., Laga, H., Picard, D., & Gosselin, P. H. (2014). Covariance descriptors for 3D shape matching and retrieval. In *2014 IEEE Conference on computer vision and pattern recognition (CVPR)* (pp. 4185–4192). IEEE.

Tombari, F., Salti, S., & Di Stefano, L. (2010). Unique shape context for 3D data description. In *Proceedings of the ACM workshop on 3D object retrieval, 3DOR '10* (pp. 57–62). New York, NY, USA: ACM. doi:10.1145/1877808.1877821.

Tombari, F., Salti, S., & Di Stefano, L. (2010). Unique signatures of histograms for local surface description. In *Computer vision–ECCV 2010* (pp. 356–369). Springer.

Tombari, F., Salti, S., & Stefano, L. D. (2011). A combined texture-shape descriptor for enhanced 3D feature matching. In *2011 18th IEEE international conference on image processing (ICIP)* (pp. 809–812). IEEE.

Tombari, F., Salti, S., & Stefano, L. D. (2013). Performance evaluation of 3D keypoint detectors. *International Journal of Computer Vision*, *102*(1–3), 198–220. doi:10.1007/s11263-012-0545-4.

Trzcinski, T., Christoudias, M., & Lepetit, V. (2015). Learning image descriptors with boosting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(3), 597–610.

Yang, X., & Cheng, K. T. (2014). Local difference binary for ultrafast and distinctive feature description. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *36*(1), 188–194.

Zhong, Y. (2009). Intrinsic shape signatures: A shape descriptor for 3D object recognition. In *2009 IEEE 12th international conference on computer vision sorkshops (ICCV workshops)* (pp. 689–696). doi:10.1109/ICCVW.2009.5457637

Zhou, W., Li, H., Hong, R., Lu, Y., & Tian, Q. (2015). BSIFT: Toward data-independent codebook for large scale image search. *IEEE Transactions on Image Processing*, *24*(3), 967–979. doi:10.1109/TIP.2015.2389624.

Sai Manoj Prakhya received B.Tech degree from Amrita University, India, in 2012. He is currently pursuing Ph.D. degree at the School of Computer Engineering, Nanyang Technological University, Singapore. He is an attached student with the Robotics Programme at the Institute for Infocomm Research, A*STAR, Singapore. His research interests include 3D perception, point cloud processing and RGB-D SLAM.

Bingbing Liu is a Research Scientist with Institute for InfoComm Research, A*STAR, Singapore and he also leads the group of Localization and Mapping of the Autonomous Vehicle Department. He received his Bachelor degree from Harbin Institute of Technology, China in 2000 and the Ph.D degree from Nanyang Technological University, Singapore in 2007. His research interests include SLAM, inertial navigation, sensor fusion for mapping and localization, and 3D Lidar based perception for autonomous vehicles etc.

Weisi Lin received the Ph.D. degree from Kings College, London University, London, U.K., in 1993. He is currently an Associate Professor with the School of Computer Engineering, Nanyang Technological University, and served as a Laboratory Head of Visual Processing, Institute for Infocomm Research. He has authored over 300 scholarly publications, holds seven patents, and receives over U.S. 4 million in research grant funding. He has maintained active long-term working relationship with a number of companies. His research interests include image processing, video compression, perceptual visual and audio modeling, computer vision, and multimedia communication. Dr. Lin served as an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE SIGNAL PROCESSING LETTERS, and Journal of Visual Communication and Image Representation. He is also on six IEEE Technical Committees and Technical Program Committees of a number of international conferences. He was the lead Guest Editor for a special issue on perceptual signal processing of the IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING in 2012. He is a Chartered Engineer in the U.K., a fellow of the Institution of Engineering Technology, and an Honorary Fellow of the Singapore Institute of Engineering Technologists. He was the Co-Chair of the IEEE Multimedia Communications Technical Committee special interest Group on Quality of Experience. He was an Elected Distinguished Lecturer of APSIPA in 2012/2013.

Vinit Jakhetiya received B.Tech degree specializing in Computer and Communication Engineering from LNM Institute of Information Technology, India in 2011. He is currently pursuing Ph.D. degree in Electronics and Computer Engineering from Hong Kong University of Science and Technology (HKUST), Hong Kong. His research interests include image interpolation, image/video coding and image/video denoising. He was the recipient of the best undergraduate paper award from the 28th IEEE International Instrumentation and Measurement Technology Conference (2011).

Sharath Chandra Guntuku received B.E. (Hons) degree from Birla Institute of Technology and Science (BITS) - Pilani, India, in 2013. He is currently pursuing Ph.D. degree at the School of Computer Engineering, Nanyang Technological University, Singapore. His research interest lies in computer vision and multimedia, specifically modeling and predicting users preferences/ behavior based on social media data.