2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Congress Center Hamburg
Sept 28 - Oct 2, 2015. Hamburg, Germany

B-SHOT : A Binary Feature Descriptor for Fast and Efficient Keypoint Matching on 3D Point Clouds

Sai Manoj Prakhya[†], Bingbing Liu[‡] and Weisi Lin[†]

Abstract— **In this paper, we introduce the very first 'binary' 3D feature descriptor, B-SHOT, for fast and efficient keypoint matching on 3D point clouds. We propose a binary quantization method that converts a real valued vector to a binary vector. We apply this method on a state-of-the-art 3D feature descriptor, SHOT [1], and create a new binary 3D feature descriptor. B-SHOT requires 32 times lesser memory for its representation while being 6 times faster in feature descriptor matching, when compared to the SHOT feature descriptor. Experimental evaluation shows that B-SHOT offers comparable keypoint matching performance to that of the state-of-the-art 3D feature descriptors on a standard benchmark dataset.**

I. INTRODUCTION

With the advent of Microsoft Kinect and Asus Xtion Pro Live sensors, 3D data acquisition has become affordable. A recent project of Google's ATAP, Project Tango, can provide online 3D pose estimates and depth data from a hand-held mobile device. With the availability of these mobile 3D data acquisition devices, there is a need to develop applications that have low memory footprint and consume less power so that they can run efficiently.

Keypoint matching is a pre-requisite step in many applications such as Simultaneous Localization and Mapping (SLAM), Sparse Depth Odometry (SDO) [2], 3D object recognition [3], 3D shape retrieval and point cloud registration [4]. Most of these applications involve 3D keypoint detection and their matching via feature descriptors to find true keypoint correspondences. Tombari et al. [5] have performed a comprehensive performance evaluation of various 3D keypoint detectors, while other prominent 3D keypoint detectors include SURE [6] and NARF [7] keypoint detectors. Once the keypoints are detected on 3D point clouds, the next step is to match them via feature descriptors. Feature descriptors essentially encode the neighbourhood around a keypoint into a multi-dimensional vector. These feature descriptors are generally matched by calculating Euclidean distance in their high dimensional vector space, which is computationally expensive. Moreover, the computational requirements for feature descriptor based matching increase accordingly with the increase in the dimension of the feature descriptor.

In order to decrease the computational power and time required for feature descriptor matching, many binary feature descriptors [8]–[10] were proposed in the 2D image domain. The first advantage of these binary descriptors is that they require much lesser memory to represent and store them. The second advantage is that binary feature descriptors can be matched extremely fast by employing Hamming distance.

In this paper, we propose the very first binary 3D feature descriptor (as per our knowledge) that describes local 3D surface neighbourhoods around a keypoint. There are binary feature descriptors for 2D images [8]–[10], but there is none for 3D local surface description. Essentially, we transform a state-of-the-art 3D feature descriptor, Signatures of Histograms of OrienTations (SHOT) [1], consisting of 352 dimensions, requiring 1408 bytes of memory, into a 352 bit binary descriptor dubbed 'B-SHOT'. There is a 32 fold decrease in the memory required for its storage. Moreover, binary feature descriptor matching can be performed extremely fast and our experiments showed 6 times faster matching of B-SHOT descriptors over SHOT descriptors.Finally, B-SHOT feature descriptor highlights the feasibility of binary 3D feature descriptors and opens up whole new possibilities of developing highly efficient binary 3D feature descriptors.

Recently, Guo et al. [3], [11] have performed an extensive survey [11] and performance evaluation [3] of various 3D local feature descriptors. We compare the proposed B-SHOT with SHOT [1], RSD [12] and FPFH [13]feature descriptors. As we create B-SHOT from SHOT and use it extensively throughout the paper, we present the principles behind the construction of the SHOT descriptor construction in the next section. FPFH [13] is an enhancement to the initially proposed Point Feature Histogram (PFH), wherein, a Darboux frame is created at every point in the local neighbourhood and 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, the distance parameter is dropped as it does not cater to point cloud density variations. Moreover, FPFH is speeded up by calculating Simple PFH and weighing them to create a 33 dimensional FPFH feature descriptor. The Radius based Surface Descriptor (RSD) [12] calculates the distance between the keypoint and all the neighbourhood points and the angle between their normals. From these, the ones with the maximum and the minimum radii are considered to create the descriptor while others are discarded.

Coming on to the binary feature descriptors in 2D image domain, some prominent ones [8], [10] are constructed by

*This research is partially supported by both the Singapore National Research Foundation under its IDM Futures Funding Initiative, administered by the Interactive & Digital Media Programme Office, Media Development Authority and Agency for Science, Technology and Research (A*STAR), Singapore.

[†]Sai Manoj Prakhya and Weisi Lin are with the School of Computer Engineering, Nanyang Technological University, Singapore. Sai Manoj Prakhya's email is `SAIMANOJ001@e.ntu.edu.sg` and Weisi Lin's email is `wslin@ntu.edu.sg`

[‡]Bingbing Liu is with the Institute for Infocomm Research, A*STAR, Singapore. His email is `bliu@i2r.a-star.edu.sg`

978-1-4799-9994-1/15/$31.00 ©2015 IEEE

comparing the intensity values of the center pixel with another pixel chosen from a sampling pattern defined around the center pixel. There are other works that employ learning [14], hashing [15], vector quantization and PCA based techniques for dimension reduction too. However, it is not straightforward to extend some of these methods to 3D point clouds as the considered point clouds can be unordered and there is no direct correspondence of intensity in 2D images to 3D point clouds. Hence, in this paper, we propose a generic approach to convert a feature descriptor into a binary representation, which can be applied to 2D feature descriptors as well.

II. SHOT Feature Descriptor

Inspired from SIFT [16] and considering the advantages of both signature and histogram based methods [1], Signatures of Histograms of OrienTations (SHOT), was proposed in [1]. In our proposed method, we transform the SHOT feature descriptor into a binary representation, B-SHOT, as explained in Section III. Hence for completeness, we briefly describe the working principles behind the SHOT feature descriptor.

To make the feature descriptor invariant to rotation, translation and scaling, the authors propose to create a local reference frame from the eigenvector of the modified neighbourhood covariance matrix $\mathbf{C}$. They 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 $\mathbf{q}$, as shown in Eqn. 1.

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

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 are representing. Finally, local $\mathbf{y}$ axis is obtained by the cross product of $\mathbf{z}$ and $\mathbf{x}$, i.e., $\mathbf{y} = \mathbf{z} \times \mathbf{x}$.

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 (only 4 azimuth partitions are shown for better visibility). This 3D spherical grid has 32 partitions arising from 8 azimuth, 2 elevation and 2 radial divisions. 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 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.

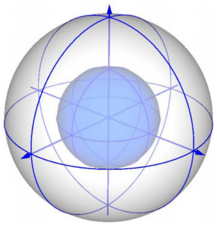


Fig. 1: Spherical grid used in SHOT descriptor [1].

To cope with the boundary effects arising from histogram based binning and small perturbations in local reference frame, quadrilinear interpolation technique is employed. Finally the descriptor is normalized to make it robust against point density variations.

III. B-SHOT Feature Descriptor

The SHOT feature descriptor is a 352 dimensional vector 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. Let us consider a SHOT descriptor S_i, where $i = \{0,1,2,..,351\}$ and each value of S_i can be any decimal value between 0 and 1. To create a B-SHOT feature descriptor, we encode the 352 dimensional SHOT descriptor into 352 bit binary descriptor. Let us represent the newly created B-SHOT feature descriptor by B_i, where $i = \{0,1,2,..,351\}$ and each value of B_i is either 0 or 1. This encoding of SHOT to B-SHOT is performed by an iterative procedure that takes four consecutive values from the beginning of S_i and encodes them into corresponding four binary values in B_i.

We consider various possibilities of the four values in S_i and encode them accordingly into B_i. Let us consider four values from a SHOT descriptor, S_i, $\{S_0,S_1,S_2,S_3\}$ and its corresponding bits to be encoded into B-SHOT descriptor, B_i, $\{B_0,B_1,B_2,B_3\}$.

Here are the various possibilities :

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

- Case A : If all the 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 is shown in Algorithm 1.
- 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 $\{1,0,1,1\}$. Pseudocode for this is shown in Algorithm 2.

Algorithm 1 Psuedocode for Case C

```
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 2 Pseudocode for Case D

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

- 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 $\{1, 1, 1, 1\}$.

In this way, the four real values from the considered SHOT feature descriptor, S_i, $\{S_0, S_1, S_2, S_3\}$ are encoded into four binary bits of $\{B_0, B_1, B_2, B_3\}$. Please note that we go through these steps in a sequential way, i.e., the possibility of the existence of a single value that amounts to 90% of S_{sum} is checked first, before moving forward to check for the existence of two values that amount to 90% of S_{sum} and so on. We perform the proposed binary encoding method on all consecutive sets of 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.

It is important to note that there is a 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 from S_i is ignored if they sum to 90% of S_{sum}. For example, let us look at this extreme case, if $\{S_0, S_1, S_2, S_3\}$ = {0.85, 0.14, 0, 0}, then the encoded $\{B_0, B_1, B_2, B_3\}$ would be {1, 1, 0, 0}. As it can be seen, the binary representation highlights that bits B_0 and B_1 are same, but in reality, they are not. As a result of this information loss, in most of the cases, slightly lower number of correspondences are found using B-SHOT when compared to the number of correspondences found using SHOT feature descriptor. Please note that it is enough if a rough 3D transformation is estimated by the established correspondences, as ICP [4], [17] algorithm can later be used to obtain a more accurate transformation estimate.

IV. Experimental Evaluation

The source code of the implementation and updates to the proposed method will be made available at *https://sites.google.com/site/bshotdescriptor/*

A. Experimental Setup

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

2) Dataset: We employ the publicly available Kinect dataset created by Tombari et al. [5] for the experimental evaluation. The Kinect dataset provides *models*, *scenes* and respective ground truth 3D transformations between them. In this dataset, *scenes* contain various objects in different orientations and occlusions whereas *models* represent the objects. 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 approximately 3 models/objects present in it and there are 49 scene-model pairs in total.

B. Experimentation Method

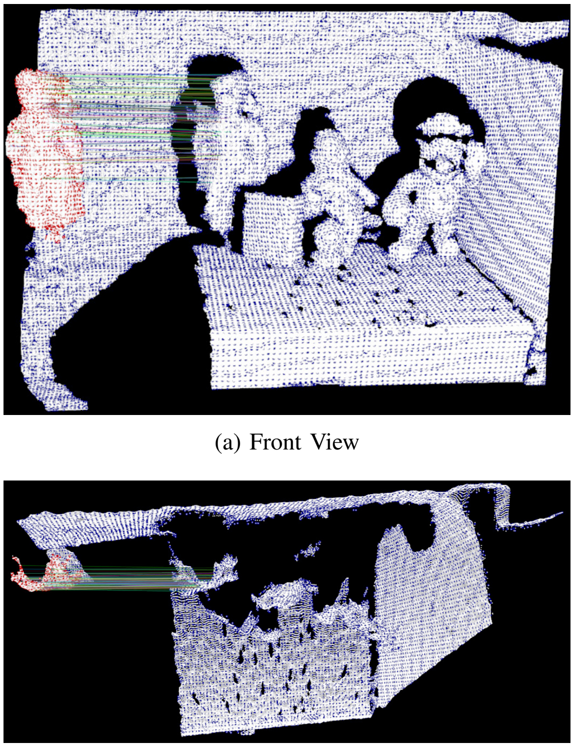
We evaluate B-SHOT and compare it with SHOT [1], RSD [12] and FPFH [13] feature descriptors as mentioned below.

- Detect uniform 3D keypoints with a voxel grid filter of radius R_{kp} on both the scene and the model. This caters for the highest possible keypoint detection ambiguity from 3D keypoint detectors and the possible false correspondences that can arise from the background present in the scene.
- Extract 3D feature descriptors with a support radius of R_{fd} around the found keypoints on both the scene and the model and establish nearest neighbour correspondences that are reciprocal.
- Apply RANSAC to find the final set of correspondences and estimate the 3D transformation T_R using the maximal consensus set.

To compute the difference, T_{diff}, between the estimated 3D transformation via RANSAC, T_R, and the available ground truth transformation, T_{GT}, we use the Euclidean metric as shown below.

$$T_{diff} = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} (T_{R_{ij}} - T_{GT_{ij}})^2} \qquad (2)$$

T_{diff} metric provides a quantitative measure of how well the established correspondences fulfil the purpose of accurately estimating the 3D transformation between the *scene*



(a) Front View

(b) Top View

Fig. 2: B-SHOT Correspondences after RANSAC on a scene and model from the Kinect dataset. It can be seen that there is no false correspondence.

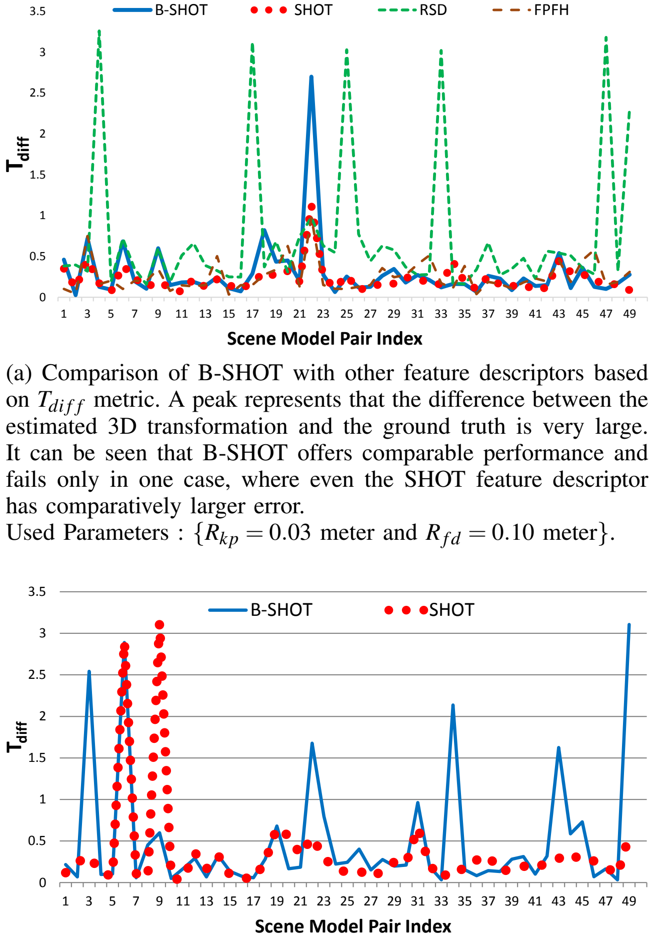
and the *model*. Once the 3D transformation is estimated, the next thing to evaluate is how many detected keypoints are positively matched via feature descriptors. To quantify this, we provide the RANSAC correspondences ratio C_{RK} as shown below.

$$C_{RK} = \frac{No.\ of\ final\ RANSAC\ correspondences}{No.\ of\ keypoints\ detected\ on\ the\ model} \quad (3)$$

C. Experimental Results

In Fig 2, we show B-SHOT correspondences after RANSAC between a scene and a model from the Kinect dataset. In the figure, uniform keypoints were detected with radius $R_{kp} = 0.01m$ and the B-SHOT descriptor support size was $R_{fd} = 0.15m$. As can be seen from the figure, there is no false correspondence. 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.

In Fig. 3, we compare B-SHOT with other feature descriptors based on the T_{diff} metric as shown in Equation 2. Fig. 3(a) compares B-SHOT with SHOT [1], RSD [12] and FPFH [13] feature descriptors. The keypoints were detected using voxel grid filter with leaf size, $R_{kp} = 0.03m$ and feature descriptors were extracted with support size of $R_{fd} = 0.10m$ on all scene-model pairs. SHOT, FPFH and RSD feature descriptors were matched based on Euclidean distance metric while Hamming distance was employed to match B-SHOT feature descriptors. Nearest neighbour reciprocal feature descriptor correspondences were established. Finally, the best consensual transformation is found using RANSAC and the T_{diff} metric is estimated. A higher value of T_{diff} indicates that the estimated 3D transformation is quite different from the ground truth. From Fig. 3(a), it can be seen that B-SHOT offers similar performance to that of the SHOT feature descriptor and fails only in one case, where even the SHOT feature descriptor had a relatively larger error. Though RSD feature descriptor had lesser computational time, its high inaccuracy has inhibited us to employ RSD in later comparisons. FPFH offered good matching performance but its computational time increases exponentially with the increase in the support radius[1]. In our next experimentation where we set the support radius, R_{fd} to $0.15m$ and with substantial increase in number of keypoints, ($R_{kp} = 0.01m$), FPFH was



(a) Comparison of B-SHOT with other feature descriptors based on T_{diff} metric. A peak represents that the difference between the estimated 3D transformation and the ground truth is very large. It can be seen that B-SHOT offers comparable performance and fails only in one case, where even the SHOT feature descriptor has comparatively larger error.
Used Parameters : $\{R_{kp} = 0.03$ meter and $R_{fd} = 0.10$ meter$\}$.

(b) B-SHOT and SHOT are compared based on T_{diff} metric. In this case, the keypoints are very close ($R_{kp} = 0.01m$) and there is high correlation between the feature descriptors of neighbouring keypoints. As the support size is also large ($R_{fd} = 0.15m$), the occlusion/unavailability of the data from the model is also another reason for a decreased performance in both the descriptors.
Used Parameters : $\{R_{kp} = 0.01$ meter and $R_{fd} = 0.15$ meter$\}$.

Fig. 3: Comparison of B-SHOT with other 3D feature descriptors based on T_{diff} metric. It can be seen that B-SHOT offers comparable performance with respect to the state-of-the-art.

[1] This can also be seen from Fig. 9 of Salti et al. [1]

consuming atleast 10-fold more time when compared to SHOT, for feature descriptor computation. Based on these two factors, RSD and FPFH feature descriptors were not considered in the following experiments.

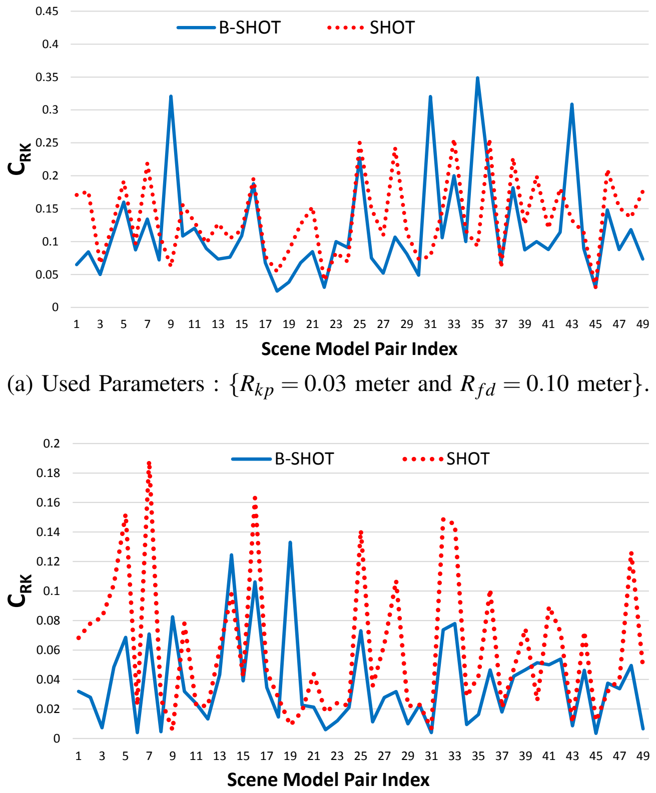
In Fig. 3(b), we compare SHOT and B-SHOT with keypoints being detected with $R_{kp} = 0.01m$ and feature descriptors are extracted with $R_{fd} = 0.15m$. In this setting, there will be a large correlation between feature descriptors of the neighbouring keypoints. B-SHOT offered very good performance in the previous scenario where the keypoints are well separated as they lie for every $0.03m$ and descriptor support radius was $0.10m$. But B-SHOT offered a slightly lower performance in this scenario as keypoints lie for every $0.01m$ and the quantization makes the feature descriptors of neighbouring keypoints ever more similar. One more reason for the degraded performance of both the feature descriptors is that for a support radius of $0.15m$, there are occlusions in some models which are not present in the scene, hence the discrepancies in feature descriptor creation may also be a reason. In the figure, there is a case where B-SHOT offered better performance than SHOT and the reason is that the the quantization/binarization performed by B-SHOT has enabled it to represent the inaccurate data in a better way and aided the feature matching process. Finally from Fig. 3(a) and Fig. 3(b), it can be said that B-SHOT can estimate the 3D transformation between two point clouds accurately without failing in large number of cases.

In our next experiment as shown in Fig. 4, we evaluate the ratio of the detected keypoints on the model that are positively matched via feature descriptors after RANSAC. This is captured by C_{RK} as shown in Equation 3. It can be seen from Fig. 4 that B-SHOT offers slightly lower ratio of keypoint matches when compared to SHOT. The reason for this being the quantization performed in order to reduce the descriptor size. It can be noted that six true correspondences are enough to find 3D transformation that can be used to trigger ICP [17] later on for accurate 3D transformation estimation. Hence, as long as the 3D transformation can be estimated accurately, as already shown and validated by Fig. 3(a), it is good enough in most of the applications.

D. Computational & Memory Requirements

One of the biggest advantages of a binary feature descriptor is that feature descriptor matching can be performed extremely fast. 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 binary XOR operation of the two input binary vectors. This can be computed even faster on CPU's that have SSE 4.1 enabled, which supports the POPCNT instruction.

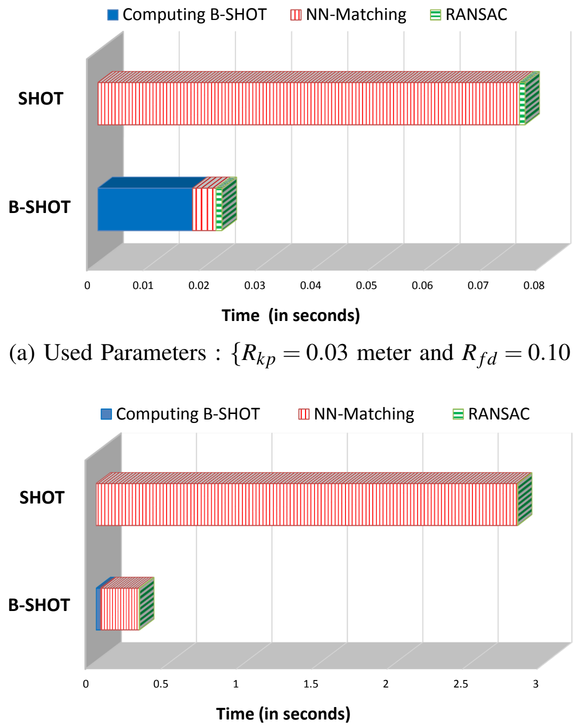
We present the computational time required for B-SHOT and SHOT descriptor based matching of keypoints in Table. I. An important point to note is that the B-SHOT descriptor is created from the SHOT descriptor and hence the extra time required to create B-SHOT from SHOT is also considered in the Table. I. However, as both B-SHOT



(a) Used Parameters : $\{R_{kp} = 0.03$ meter and $R_{fd} = 0.10$ meter$\}$.

(b) Used Parameters : $\{R_{kp} = 0.01$ meter and $R_{fd} = 0.15$ meter$\}$.

Fig. 4: Comparison of B-SHOT with SHOT feature descriptor based on C_{RK} metric as shown in Equation 3. A higher value represents greater number of keypoint matches after RANSAC with respect to detected keypoints on the model.



(a) Used Parameters : $\{R_{kp} = 0.03$ meter and $R_{fd} = 0.10$ meter$\}$.

(b) Used Parameters : $\{R_{kp} = 0.01$ meter and $R_{fd} = 0.15$ meter$\}$.

Fig. 5: Computational time (in seconds) for matching B-SHOT and SHOT feature descriptor on the Kinect dataset.

TABLE I: Computational time (in seconds) required for matching B-SHOT and SHOT feature descriptors

	Computing B-SHOT	*NN-Matching*	*RANSAC*
Case A			
B-SHOT	0.0168	0.0041	0.0010
SHOT	(Not Applicable)	0.0751	0.0010
Case B			
B-SHOT	0.0308	0.2554	0.0018
SHOT	(Not Applicable)	2.7977	0.0011

and SHOT feature descriptors have the same offset computation time for calculating SHOT feature descriptors, we do not show them as it would overshadow the enhancement in feature descriptor matching offered by B-SHOT feature descriptor when compared to SHOT. The time taken for computing B-SHOT from SHOT, matching B-SHOT and finding the final correspondences via RANSAC are all computed and averaged over the 49 scene-model pairs. Just to re-iterate, NN-matching of B-SHOT feature descriptors is obtained by employing hamming distance metric and discarding the non-reciprocal correspondences. NN-matching of SHOT feature descriptors are obtained by creating a kdtree representation and retrieving only the reciprocal correspondences[2].

A graphical representation is shown in Fig. 5, while Table. I shows the exact values that were acquired during our experimentation. In Table 5, *Case A* refers to the experimental settings where $R_{kp} = 0.03m$ and $R_{fd} = 0.10m$ (as shown in Fig. 5 (a)), while *Case B* refers to the experimental settings where $R_{kp} = 0.01m$ and $R_{fd} = 0.15m$ (as shown in Fig. 5(b)). In *Case A*, B-SHOT feature descriptor computation and matching is approximately 3.5 times faster than SHOT descriptor matching, whereas in *Case B*, B-SHOT is 9.7 times faster than SHOT descriptor matching. This is because, in the *Case B*, there are more number of keypoints and hence SHOT descriptor matching was even more computationally intensive when compared to *Case A*. From these experiments, we can conclude that B-SHOT feature descriptor matching is approximately 6 times faster than SHOT descriptor matching.

As mentioned earlier, binary descriptors have an edge over conventional feature descriptors in terms of the memory required to store and represent them. As SHOT descriptor is of 352 dimensions, it requires 1408 bytes (in accordance with IEEE 754 single-precision binary floating-point format), 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 involve online transfer of feature descriptors.

V. Acknowledgement

We would like to thank Vinit Jakhetiya, PhD student at HKUST, Sharath Chandra Guntuku and Anh Trung Tra, PhD students at NTU, for their help while developing this work.

[2] 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.

VI. Conclusion

In this paper, we introduce the very first binary 3D feature descriptor, B-SHOT, for keypoint matching on 3D point clouds. A binary quantization method is proposed and is applied onto a state-of-the-art 3D feature descriptor, SHOT [1], to create a binary 3D feature descriptor, B-SHOT. Our experiments show that 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 of creating efficient binary 3D feature descriptors.

References

[1] S. Salti, F. Tombari, and L. Di Stefano, “SHOT: Unique Signatures of Histograms for Surface and Texture Description,” *Computer Vision and Image Understanding*, vol. 125, pp. 251–264, 2014.

[2] S. M. Prakhya, B. Liu, W. Lin, and U. Qayyum, “Sparse Depth Odometry : 3D Keypoint based Pose Estimation from Dense Depth Data ,” in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, May 2015.

[3] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, and J. Wan, “3D Object Recognition in Cluttered Scenes with Local Surface Features: A Survey,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, Nov 2014.

[4] S. M. Prakhya, B. Liu, R. Yan, and W. Lin, “ A Closed-form Estimate of 3D ICP Covariance,” in *Machine Vision Applications (MVA), The 14th IAPR Conference on*, May 2015.

[5] F. Tombari, S. Salti, and L. DiStefano, “Performance Evaluation of 3D Keypoint Detectors,” *International Journal of Computer Vision*, vol. 102, no. 1-3, pp. 198–220, 2013.

[6] T. Fiolka, J. Stückler, D. A. Klein, D. Schulz, and S. Behnke, “SURE: Surface Entropy for Distinctive 3D Features,” in *Spatial Cognition VIII*. Springer, 2012, pp. 74–93.

[7] B. Steder, R. Rusu, K. Konolige, and W. Burgard, “Point Feature Extraction on 3D Range Scans Taking into Account Object Boundaries,” in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, May 2011, pp. 2601–2608.

[8] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua, “BRIEF: Computing a Local Binary Descriptor Very Fast,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 7, pp. 1281–1298, July 2012.

[9] A. T. Tra, W. Lin, and A. Kot, “Dominant SIFT : A Novel Compact Descriptor,” in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, May 2015.

[10] S. Leutenegger, M. Chli, and R. Y. Siegwart, “BRISK: Binary Robust Invariant Scalable Keypoints,” in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2548–2555.

[11] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, and N. Kwok, “A Comprehensive Performance Evaluation of 3D Local Feature Descriptors,” *International Journal of Computer Vision*, 2015.

[12] Z.-C. Marton, D. Pangercic, N. Blodow, J. Kleinehellefort, and M. Beetz, “General 3d modelling of novel objects from a single view,” in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 3700–3705.

[13] R. B. Rusu, N. Blodow, and M. Beetz, “Fast Point Feature Histograms (FPFH) for 3D Registration,” in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2009.

[14] T. Trzcinski, M. Christoudias, and V. Lepetit, “Learning Image Descriptors with Boosting,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 37, no. 3, pp. 597–610, 2015.

[15] C. Strecha, A. M. Bronstein, M. M. Bronstein, and P. Fua, “LDAHash: Improved Matching with Smaller Descriptors,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 1, pp. 66–78, 2012.

[16] D. G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[17] P. J. Besl and H. D. McKay, “A Method for Registration of 3-D shapes.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.