3DHoPD: A Fast Low-Dimensional 3-D Descriptor

Sai Manoj Prakhya, Jie Lin, Vijay Chandrasekhar, Weisi Lin, and Bingbing Liu

***Abstract*—Three-dimensional feature descriptors are heavily employed in various 3-D perception applications to find keypoint correspondences between two point clouds. The availability of mobile devices equipped with depth sensors compels the developed applications to be both memory and computationally efficient. Toward this, in this letter, we present 3DHoPD, a new low-dimensional 3-D feature descriptor that is extremely fast to compute. The novelty lies in compactly encoding the "3-D" keypoint position by transforming it to a new 3-D space, where the keypoints arising from similar 3-D surface patches lie close to each other. Then, we propose histograms of point distributions (HoPD) to capture the neighborhood structure, thus forming 3DHoPD (3D+HoPD). We propose a tailored feature descriptor matching technique, wherein the "3-D" keypoint position in the new 3-D space is used to remove false positive matches, effectively reducing the search space by 90%, and then, the exact match is found using the "HoPD" descriptor. Experimental evaluation on multiple publicly available datasets shows that 3DHoPD is robust to noise and offers stable and competitive keypoint matching performance to the existing state-of-the-art 3-D descriptors with similar dimensionality across datasets, while requiring dramatically low-computational time (10× faster). The source code and additional experimental results are available at https://sites.google.com/site/3dhopd/**

***Index Terms*— Descriptor matching, RGB-D perception, SLAM, 3D feature descriptors, 3D object detection.**

I. Introduction

3D feature descriptors are the first resort to find correspondences between two arbitrarily oriented 3D point clouds. 3D feature descriptors are heavily employed in numerous applications spanning the domains of computer vision and robotics. With the advent of mobile devices equipped with depth sensors, such as Google Tango and affordable depth cameras such as Intel RealSense Camera and Kinect-style cameras, there is a need to develop applications that are computationally efficient and have low memory footprint. This would enable and extend various 2D image based applications to 3D offering higher accuracy and robustness, specially in the case of absence of texture and varied lighting conditions of the target environment. The first steps of many applications such as simultaneous localization and mapping [1], [2], 3D object modelling [3], 3D object recognition [4], [5], partial 3D object retrieval [6], [7] and point cloud registration [8], [9] is to extract 3D keypoints on a source and a target point cloud, match them via 3D feature descriptors and estimate the relative 3D transformation between them. And most importantly, 3D feature descriptor extraction is one of the most computationally demanding and high memory consuming steps.

Feature descriptors essentially encode the characteristics of point distributions around the keypoints into a multidimensional vector. Most of the existing 3D feature descriptors can be classified into two classes [10], namely Spatial Distribution Histograms (SDH) and Geometric Attribute Histograms (GAH). SDH based descriptors [11]–[15] represent the spatial distributions of 3D points around the detected keypoints into a multidimensional vector while GAH based descriptors [16]–[19] encode the geometric information, such as normals and curvature, calculated from the neighbourhood of the detected keypoints.

3D Shape Contexts (3DSC) [12] superimpose a spherical grid aligned with the surface normal at the considered keypoint and calculate a weighted sum of points that fall in each partition along radial, azimuth and elevation directions. Unique Shape Contexts (USC) [13] was proposed as an extension to 3DSC, wherein a unique and repeatable local reference frame is employed to superimpose the 3D spherical grid rather than a surface normal. Signature of Histograms of normal OrienTations (SHOT) descriptor tries to mimic SIFT [20] descriptor from 2D image domain and hence employs a histogram of gradients (surface normals in 3D domain) to represent the point distributions in each partition of the superimposed 3D spherical grid. FPFH descriptor is proposed as an improved version of PFH [17], which is based on Darbaux frame calculated between the keypoint and its neighbours. In FPFH, firstly Simplified PFH's (SPFH) are computed between the keypoint and each of its neighbours and then the descriptor is constructed as a weighted sum of SPFH's. RoPS [14] descriptor is constructed by accumulating various statistics calculated from the distribution matrices that are constructed by rotating the 3D surface patch along x, y, z axes and projecting it onto yz, zx, xy planes. However, RoPS requires mesh connectivity information for its computation and hence it does not readily work on raw point clouds that have $[x, y, z]$ information alone without any post-processing steps.

While the memory footprint and matching complexity reciprocate with descriptor's dimensionality, the extraction

Manuscript received October 15, 2016; accepted January 23, 2017. Date of publication February 13, 2017; date of current version March 10, 2017. This letter was recommended for publication by Associate Editor J. Miura and Editor F. Chaumette upon evaluation of the reviewers' comments. This work was supported by Nanyang Technological University and the Institute for Infocomm Research, A*STAR.

S. M. Prakhya and W. Lin are with the School of Computer Science and Engineering, Nanyang Technological University, 639798, Singapore (e-mail: saimanoj001@ntu.edu.sg; wslin@ntu.edu.sg).

J. Lin, V. Chandrasekhar, and B. Liu are with the Institute for Infocomm Research, A*STAR, 138632, Singapore (e-mail: lin-j@i2r.a-star.edu.sg; vijay@i2r.a-star.edu.sg; bliu@i2r.a-star.edu.sg).

Color versions of one or more of the figures in this letter are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LRA.2017.2667721

complexity depends on the hand crafted design of the descriptor. Though FPFH has only 33 dimensions, its extraction time is significantly higher compared to others because the effective support size for descriptor extraction becomes twice for computing SPFH [18]. With this, as the support size increases, the number of points in 3D volume increase exponentially, hence making FPFH computationally expensive in high density point clouds or with larger support size [10]. On the other hand, USC and 3DSC, with a dimensionality of 1980, employ an excessively high number of bins and represent each bin, only with a single value, hence they lose on performance and have high computational complexity. In the case of SHOT descriptor with 352 dimensions, surface normal computation takes up most part of the computational power, however, their idea of imitating SIFT by representing each grid with a histogram (not with a single value as employed in 3DSC and USC) of normals pays off with better performance on point clouds that have small surface variations.

SHOT (352 dim) and USC (1980 dim) are relatively faster in descriptor extraction compared to FPFH but demand higher memory footprint and matching complexity because of their dimensionality. Even if PCA based dimensionality reduction is employed, it would still be necessary to extract those high dimensional descriptors in the firstplace on the target mobile devices and store the PCA transformation matrices, thus resulting in high memory and computational footprints. Therefore, in this work, we propose 3DHoPD, a low dimensional (18-dim) 3D descriptor that is extremely fast to compute and offers stable and competitive performance. The novelty lies in the design of 3DHoPD (3D+HoPD) involving two steps:

1. *3D:* We propose to transform the 3D keypoints around which the descriptors are computed, into a new 3D space where keypoints arising from similar 3D surfaces lie close to each other.

Hence, in this new 3D space, for a given source keypoint, a list of probable keypoint matches with high recall can be retrieved with a simple radial search. This reduces the search space by 90% by removing the false positive matches and aids in descriptor matching.

2. *HoPD:* From this set of highly probable keypoint matches, the exact match is then found by proposing a fast 3D descriptor, HoPD (Histogram of Point Distributions).

II. 3DHoPD: The Proposed 3D Feature Descriptor

Lets consider an input source point cloud P_{source} on which N keypoints, K_n, where $n = \{1, 2, ..., N\}$ are detected using a 3D keypoint detector. Then, for each of these source keypoints, K_n, a 3DHoPD feature descriptor is created from its neighbourhood surface $Surface_{\text{nm}}$ containing M points, where $m = \{1, 2, ..., M\}$. The neighbourhood surface or the 3D surface patch around the keypoint is determined by the support size used to construct the 3D descriptor. Notation example: A keypoint K_i has a $Surface_{\text{im}}$, where i represents the keypoint index and m represents the points in the 3D surface patch around the keypoint K_i.

A. Construction

The 18 dimensional 3DHoPD descriptor encodes the 3D keypoint position in the first three dimensions and HoPD in the next 15 dimensions.

1) Encoding the 3D Keypoint Position: The considered 3D keypoint is transformed to a new 3D space and its new 3D coordinates are stored in the first three dimensions of the 3DHoPD descriptor. The task of extracting feature descriptors around keypoints and matching them to find correspondences can be seen as trying to find exact 3D keypoint correspondences, where each keypoint correspondence arises from similar 3D surface patch/neighbourhood. Hence, we transform the keypoints to a new 3D space where the keypoints arising from similar 3D surfaces lie close to each other. To achieve this, let us consider a 3D surface patch $Surface_{\text{im}}$ with m neighbourhood points around the keypoint K_i. First, we find the mean point $mean_{\text{pt}}$ or in other words, the centroid of all the points in the considered 3D surface patch $Surface_{\text{im}}$. Second, we subtract the found $mean_{\text{pt}}$ from all the 3D points in $Surface_{\text{im}}$, effectively creating a new 3D surface patch $Surface_{\text{im}-\text{mean}_{\text{pt}}}$. Third, we also subtract the found $mean_{\text{pt}}$ from the considered keypoint K_i as well, resulting in $K_{i-\text{mean}_{\text{pt}}}$. Finally, we estimate the local reference frame $[\mathbf{RF}]_{3\times3}$, as explained later, from this new 3D surface patch $Surface_{\text{im}-\text{mean}_{\text{pt}}}$, and then transform the keypoint $K_{i-\text{mean}_{\text{pt}}}$ to a new 3D space as shown below:

$$[K_{\text{iRF}}]_{3\times1} = [\mathbf{RF}]_{3\times3}[K_{i-\text{mean}_{\text{pt}}}]_{3\times1} \tag{1}$$

where K_{iRF} represents the new 3D coordinates (in new 3D space) of the considered keypoint K_i arising from a 3D surface patch $Surface_{\text{im}}$. And, $[K_{\text{iRF}}]_{3\times1}$ forms the first three dimensions of the constructed 3DHoPD descriptor.

Local Reference Frame: It is estimated from the eigenvectors of the modified covariance matrix $\mathbf{C}$ calculated from the points in the 3D surface patch $Surface_{\text{im}-\text{mean}_{\text{pt}}}$, as shown in Eqn. (2). Similar to the local reference frame calculated in SHOT [19], the 3D points, $Surface_{\text{im}-\text{mean}_{\text{pt}}}$, where $m = \{1, 2, ..., M\}$, that lie in the neighbourhood defined by support radius $\mathbf{r}$, are weighed based on their distance from the considered keypoint $K_{i-\text{mean}_{\text{pt}}}$ as shown in Eqn. (2). For readability, we represent M points in 3D surface $Surface_{\text{im}-\text{mean}_{\text{pt}}}$ by $\mathbf{q_m}$ and $K_{i-\text{mean}_{\text{pt}}}$ by $\mathbf{q}$.

$$\mathbf{C} = \frac{1}{\sum_{m:d_m \leq \mathbf{r}}(\mathbf{r} - d_m)} \sum_{i:d_m \leq \mathbf{r}} (\mathbf{r} - d_m)(\mathbf{q_m} - \mathbf{q})(\mathbf{q_m} - \mathbf{q})^{\mathbf{T}} \tag{2}$$

where $d_m = ||\mathbf{q_m} - \mathbf{q}||_2$. 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 [21]. 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 reference frame is defined as $[\mathbf{RF}]_{3\times3} = \left[\mathbf{x}^T \mathbf{y}^T \mathbf{z}^T\right]^T$.

Intuition: In the first step of encoding the 3D keypoint position, we transform the 3D keypoints arising from similar 3D surface patches into a new space where they lie in vicinity of each other. To achieve this, we remove the mean point (geometric offset) or the centroid $(mean_{\text{pt}})$ of local surface patch

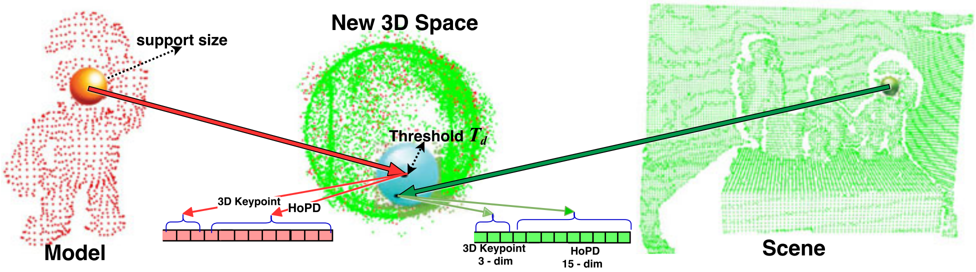


Fig. 1. The uniform model keypoints are shown in red while the scene keypoints are shown in green. The red and green spheres represent the support size for LRF and descriptor construction. There is one to one correspondence between keypoints in original space and the new 3D space. The model and scene keypoints are transformed to a new 3D space with the local reference frame. Then, for every model keypoint, a list of scene keypoints are retrieved with a radial search of radius T_d (shown as cyan colored sphere in new 3D space). From this retrieved list of probable scene keypoint matches, the one with the closest HoPD descriptor is considered as the exact match.

from all its points and then transform the keypoint to a new 3D space using the local reference frame. This local reference frame $[\mathbf{RF}]_{3\times3}$ estimated from the covariance matrix approximately captures the local 3D surface pattern. It cannot be said that the closest point in the new 3D space corresponds to the true feature match because of various practical abnormalities, such as keypoint detection ambiguity [22], noisy sensor data and the inability of the local reference frame to accurately capture the variations in the local surface. However, the true match lies in its vicinity, hence drastically reducing the search space and resulting in a small set of points to find the exact keypoint match from.

In Fig. 1, we show uniformly extracted model keypoints in red to the left, and uniform scene keypoints in green to the right. The red and green spheres on model and scene represent the support size used to construct the local reference frame and the HoPD descriptors. These model and scene keypoints are then transformed into a new 3D space that spans the volume of a unit 3D sphere, by removing the offset and multiplying them with a normalized local reference frame. There is a one to one correspondence between every model keypoint in their original space and the new 3D space, and similarly between every scene keypoint in their original space and the new 3D space. Now, to find a match for the considered model keypoint in this new 3D space, we perform a radial search with a radius of T_d as shown by a cyan colored sphere and retrieve a list of probable matches. From this list of probable matches, the one whose HoPD descriptor is closest to the considered model keypoint's HoPD descriptor, is considered as a match. This proposed 3D keypoint transformation method drastically reduces the search space, removes about 90% of false positives as shown later, and makes the job of descriptors easier, thus enabling the design of fast descriptors that offer good performance.

Rotational Invariance : This is achieved by transforming the 3D surface patch around the considered 3D keypoint using the already estimated local reference frame $[\mathbf{RF}]_{3\times3}$.

2) Histogram of Point Distributions (HoPD): The HoPD descriptor encapsulates the actual point distributions in the neighbourhood of the considered 3D keypoint into histogram. Following the same notation as in Sec. II-A, the 3D surface patch around a keypoint K_i from the source point cloud would be $Surface_{\mathrm{im-mean_{pt}}}$ with m neighbourhood points in the support region and $mean_{\mathrm{pt}}$ being the centroid. This 3D surface patch is transformed with the estimated local reference frame $[\mathbf{RF}]_{3\times3}$ to achieve rotational invariance. To construct HoPD, a D-dimensional histogram is created for x axis, by finding the minimum and the maximum x coordinate value of the points in the 3D surface patch. Then the found range between the minimum and the maximum is divided into D-bins and a D-dimensional histogram is constructed by counting the number of points that fall into each bin. The same procedure is repeated for y and z axes too, effectively creating a $3\times D$-dimensional HoPD descriptor. Our experiments have shown that $D = 5$ offers a good trade-off between performance and dimensionality, hence resulting in a 15-dimensional HoPD descriptor. The advantage of the HoPD descriptor is that its extremely fast to compute, as it does not require any computation of surface normals.

The idea is to first generate a list of highly probable matches based on the '3D' keypoint position encoded in the first three dimensions and then find an exact match with the help of fast 15-dim 'HoPD' descriptor.

B. Matching

Capitalizing on the fact that the 3D keypoints arising from similar 3D patches lie close to each other in the new 3D space, we perform the matching of 3DHoPD descriptors as follows. Firstly, the keypoints extracted on the source and the target point clouds are transformed into the new 3D space. Then, for every source 3D keypoint $K_{\mathrm{iRFsource}}$ in the new 3D space, we find a list of nearest target 3D keypoints $K_{\mathrm{iRFtarget}}$ in the new 3D space that lie within a distance threshold T_d. We perform a radial nearest neighbour search with radius T_d by employing KdTree for fast retrieval. Hence, we use the first three dimensions of 3DHoPD descriptor to generate the list of probable matches for every source keypoint. Then, in the next step, we find the exact match for the considered source keypoint from the above generated list of probable matches by employing Euclidean metric to find the exact nearest neighbour in the 15-dim HoPD descriptor's space.

III. Experimental Evaluation

We perform extensive evaluation on four publicly available datasets [10] namely, Retrieval, Queens, Random Views and Kinect, based on widely used precision-recall curves [10], [14], [19], [23] for descriptor matching experiments. These datasets

TABLE I
THIS TABLE PRESENTS AVERAGE NUMBER OF (I) MODEL KEYPOINTS, (II) SCENE KEYPOINTS, (III) THE AVERAGE SIZE OF THE LIST OF PROBABLE SCENE KEYPOINT MATCHES RETRIEVED FOR EVERY MODEL KEYPOINT (BASED ON THE "3D" KEYPOINT TRANSFORMATION IN THE NEW 3D SPACE), AND (IV) THE PERCENTAGE OF THE RETRIEVED LISTS THAT CONTAIN THE GROUNDTRUTH KEYPOINT MATCH, IN EACH OF THE DATASETS

Dataset	Model Keypoints Avg.	Scene Keypoints Avg.	Avg. size of the retrieved list (% of Scene Keypoints retrieved)	% of retrieved lists with groundtruth
Threshold: $T_d = 0.005$				
Retrieval	63	63	5 –> (7.9% of scene keypoints)	99.28%
Random Views	33	99	4 –> (4.04% of scene keypoints)	22.5%
Queens	175	862	28 –> (3.24% of scene keypoints)	14.7%
Kinect	88	1254	48 –> (3.82% of scene keypoints)	35.61%
Threshold: $T_d = 0.0075$				
Retrieval	63	63	9 –> (14.28% of scene keypoints)	99.61%
Random Views	33	99	10 –> (10.10% of scene keypoints)	34.08%
Queens	175	862	81 –> (9.39% of scene keypoints)	28.01%
Kinect	88	1254	113 –> (9.01% of scene keypoints)	50.40%
Threshold: $T_d = 0.02$				
Retrieval	63	63	31 –> (49.2% of scene keypoints)	99.76%
Random Views	33	99	50 –> (50.5% of scene keypoints)	77.8%
Queens	175	862	559 –> (64.84% of scene keypoints)	78.8%
Kinect	88	1254	611 –> (48.72% of scene keypoints)	84.28%

come with models, scenes and the groundtruth 3D transformation between them. The term 'model' represents an object point cloud, while the term 'scene' represents a point cloud containing various objects in different orientations, noise, occlusion and clutter.

A. Evaluation of the 3D Keypoint Transformation Technique

The main contribution of this work is to transform 3D keypoints to a new 3D space, based on the local reference frame, so that the keypoints arising from similar 3D surface patches lie close to each other. We perform the following experiment with the steps given below to evaluate and validate the proposed concept on all four datasets, whose point cloud data comes arises from various sensing modalities.

1) First, ISS (Intrinsic Shape Signatures) [24] keypoints with parameters as mentioned later, are detected on a scene and they are transformed on to the model based on the available groundtruth transformation, hence the groundtruth 3D keypoint correspondences are known.
2) Second, the model keypoints and the scene keypoints are transformed into the new 3D space by multiplying with the estimated LRF as mentioned in Section II-A.
3) Third, in this new 3D space, for every model keypoint, a radial search is performed with a radius of T_d and a list of nearest neighbours, i.e., list of probable scene keypoint matches is retrieved.
4) Finally, we check, if the list of scene keypoint matches that are retrieved for every model keypoint contains the actual groundtruth scene keypoint match or not.

The results are presented in Table I and for each dataset, the table shows the average number of (i) model keypoints, (ii) scene keypoints, (iii) the average size of the retrieved list of scene keypoints in the new 3D space for every model keypoint, and finally (iv) the percentage of retrieved lists that contain the groundtruth correspondence in them. In the fourth column of Table I, we also present the size of the retrieved list in terms of the percentage of scene keypoints, so that it gives an idea of how many false positives are removed. We run the experiments with three different radial thresholds, $T_d = \{0.005, 0.0075, 0.02\}$ that determines the number of the nearest neighbours retrieved in the new 3D space. In all the experiments, the support size of descriptors is set to 0.06 m, which is the same for local reference frame computation as well. In the Retrieval dataset, the models and the scenes are essentially the same, which is the reason for having same number of model and scene keypoints in Table I. However, scenes contain Gaussian noise of $\sigma = 0.1mr$ (mesh resolution) and scene-model pairs are highly rotated in this dataset.

It can be seen from Table I that, for each dataset, as T_d increases, the size of the retrieved list and the $\%$ of retrieved lists with groundtruth also increases. As the size of the retrieved list increases, many false positives creep in and a highly descriptive descriptor is required to filter the false matches and find the exact true keypoint match. If the retrieved list is small, which is achievable with small T_d, the $\%$ of retrieved lists with groundtruth also decrease. Hence, it is ideal to choose a reasonable value that gives a fair size of retrieved list and in all the descriptor matching and evaluation experiments carried out later on, we employed $T_d = 0.0075$, as it offered good trade-off between the size of the retrieved list and the $\%$ of groundtruth matches it contains.

For 3D keypoint matching, each model keypoint descriptor has to be matched/compared against each and every scene keypoint descriptor to find the final exact match. The biggest advantage with the proposed 3D keypoint transformation is that, by using just three dimensions of the descriptor which is the 3D keypoint location, a small list of highly probable source keypoint matches can be retrieved and hence, a simple and fast descriptor can be designed to find the exact match. It can be seen from Table I that, by employing a threshold $T_d = 0.0075$, approximately only 10% of the source keypoints need to be searched to find an exact match for every model keypoint and effectively reduces the search space by 90%. This proposed '3D' keypoint transformation to a new 3D space greatly removes the false positives and makes the job of descriptors easier. Our experiments have shown the practical effectiveness of setting the value of T_d to 0.0075. Hence, it can be said that, the threshold T_d can be chosen in such a way that the size of retrieved list approximately amounts to 10% of the scene keypoints. This reduces the search space by 90%, removes false positive and eases the job of the 3D descriptor. Moreover another direct implication from reducing the search space by 90% is the reduction of the computational time required for descriptor matching.

It may seem from Table I that at $T_d = 0.0075$, the $\%$ of retrieved lists with groundtruth for Random Views and Queens datasets is lower, however, the results of descriptor evaluation, as shown later, highlight that only a few matches are found even by the state-of-the-art descriptors. For example, on the Queens dataset, with $T_d = 0.0075$, the % of retrieved lists with groundtruth is 28%, which means that with an ideal descriptor, the highest achievable recall would be 0.28. However, the best recall achieved by the state-of-the-art/proposed descriptor was just ≈ 0.04, as shown in Fig. 4. This means that, there is lot of scope for improvement in the design of descriptors.

Another reason behind Queens and other datasets having relatively less $\%$ of retrieved lists with groundtruth is that, the detected keypoints can lie on planar or slightly curved regions of the point clouds. In such scenarios, it is hard to capture the 3D surface characteristics as it is not constrained in x, y, z directions. Moreover, Queens dataset has noisy point cloud data with low resolution and the inability of the local reference frame to accurately characterize or capture the nature of 3D surface patch is another reason that effects the % of the retrieved lists that contain groundtruth. The performance of the 3D descriptors varies with the nature of the point cloud data as can be seen by the distinctly varying performance on various datasets.

B. 3DHoPD Descriptor Evaluation

Precision-Recall Curves: Firstly, a set of keypoints are detected on a scene and those detected scene keypoints are transformed onto the model with the available groundtruth. In this way, all the possible false positives that may arise from the background and clutter from the scene are also accounted for. Next, we extract the considered 3D feature descriptors at these 3D keypoints on both the scene and the model. For every model keypoint's descriptor, we find the first and the second nearest scene keypoint's descriptor based on Euclidean distance metric. If the ratio of the Euclidean distance of first to the second nearest neighbour is less than or equal to a threshold α, then it is considered as a correspondence C_r. The threshold α is varied from 0 to 1 and accordingly the number of correspondences $C_{r\alpha}$ on the considered scene-model pair at various values of α is estimated. Specifically, we set the values of $\alpha = \{0.2, 0.4, 0.6, 0.75, 0.85, 0.925, 0.95, 0.975, 1.0\}$ and calculate the correspondences $C_{r\alpha}$. The groundtruth matches G_m is equal to the number of model keypoints because, for every model keypoint there lies a corresponding scene keypoint, which can be found via the available groundtruth. The true correspondences $True_{\text{corr}}$ are the correspondences from the set of $C_{r\alpha}$ that align with the groundtruth. Specifically, a correspondence in $C_{r\alpha}$ is considered as a true correspondence T_{corr}, if the matched scene keypoint lies within a threshold ϵ to the transformed model keypoint based on the groundtruth. Finally precision and recall at a specific α is calculated as:

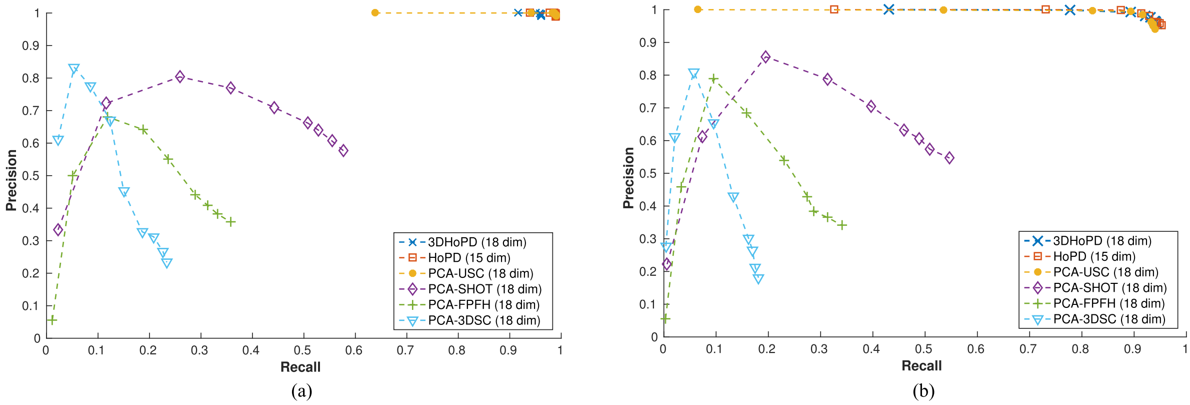
$$Precision = \frac{True_{\text{corr}}}{C_{r\alpha}} \qquad Recall = \frac{True_{\text{corr}}}{G_m} \tag{3}$$

where G_m is the number the groundtruth matches, $True_{\text{corr}}$ represents true correspondences and $C_{r\alpha}$ represents the correspondences that pass the nearest neighbour based ratio test at the considered α. Then, the precision and the recall values at a specific α is the averaged value estimated over all scene-model pairs in the considered dataset.

ISS keypoint detector was employed for keypoint detection as it offered good keypoint repeatability [22] and efficient keypoint matching performance when combined with various 3D feature descriptors [10]. All the considered 3D feature descriptors are extracted with a support size of 0.06 m around the detected ISS keypoints. In the case of 3DHoPD, T_d as mentioned in Section II-B is set to 0.0075 m. The following are the parameters used for ISS keypoint detection which is publicly available from PCL [8]: model resolution $mr = 0.0015$ m, $iss\_gamma\_21 = iss\_gamma\_32 = 0.8$, $salient\_radius = 10 \times mr$, $non\_max\_radius = 6 \times mr$, $normal\_radius = 6 \times mr$ and $border\_radius = 2 \times mr$.

Results: The proposed 3DHoPD is compared against state-of-the-art descriptors, SHOT [19], USC [13], FPFH [18] and 3DSC [12]. Each descriptor has varied dimensionality (memory footprint) and accordingly require varied computational time for descriptor matching at full dimensional settings. As the proposed 3DHoPD is significantly lower in dimensionality and in order to have a fair comparison, we create low dimensional representations of the considered SHOT, FPFH, USC and 3DSC descriptors using Principal Component Analysis (PCA) as described in our previous work [25]. In order to learn the PCA transformation matrix for a specific 3D descriptor, we extract 100,000 descriptors from the dataset[1] that has point clouds from various indoor environments and perform PCA on them. These learnt PCA transformation matrices, which are made publicly available [25], are then used to create 18-dim variants represented as PCA-SHOT, PCA-FPFH, PCA-USC and PCA-3DSC descriptors. Our previous work [25] shows that, after PCA based dimensionality reduction, about 70% of SHOT descriptors's performance is retained in the first 20 dimensions while 95% of FPFH descriptor's performance is retained in the first 20 dimensions, of their low dimensional representations. We also present the performance of 15-dim HoPD descriptor to show the enhancement offered by the proposed '3D' keypoint transformation. The experimental results, where 3DHoPD is compared with full dimensional 3D descriptors, SHOT, USC, FPFH and 3DSC, and the proposed 3D keypoint transformation technique being applied to these descriptors is available at *https://sites.google.com/site/3dhopd/*

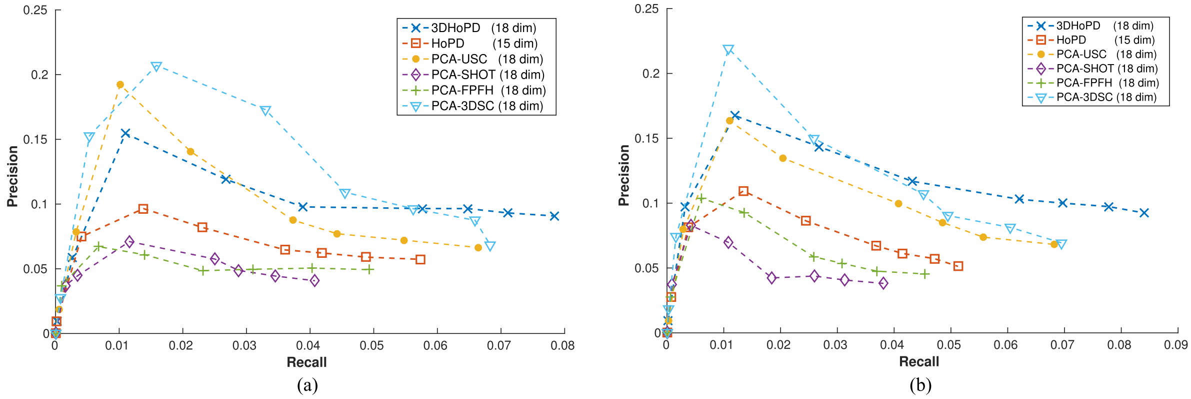
[1] https://sites.google.com/site/3dkeypoints/



(a)

(b)

Fig. 2. Performance on Retrieval dataset with varying amount of Gaussian noise with standard deviation $\sigma = \{0.1, 0.5\}\ mr$. (a) Retrieval dataset with Gaussian noise of $\sigma = 0.1mr$. (b) Retrieval dataset with Gaussian noise of $\sigma = 0.5mr$.



(a)

(b)

Fig. 3. Performance on Random Views dataset with Gaussian noise of standard deviation $\sigma = \{0.1, 0.5\}\ mr$. (a) Random Views dataset with Gaussian noise of $\sigma = 0.1mr$. (b) Random Views dataset with Gaussian noise of $\sigma = 0.5mr$.

Performance on Retrieval dataset: Here, we evaluate the performance of 3DHoPD and the considered state-of-the-art 3D descriptors on Retrieval dataset with varying amount of Gaussian noise added on to them. In this dataset, the model and the scene point clouds are the same, however, the scene point clouds are corrupted with noise. Fig. 2(a) shows the performance when Gaussian noise with a standard deviation of $0.1\ mr$ (mesh resolution) is added while in Fig. 2(b), Gaussian noise with $\sigma = 0.5mr$ is added. It can be seen from Fig. 2(a) and (b) that HoPD, 3DHoPD and PCA-USC offer state-of-the-art performance. Secondly, a closer look at Fig. 2(b) shows that 3DHoPD offers greater precision and recall at a particular α when compared to PCA-USC (18 dim). Please note that the first marker on each precision-recall curve in the figure refers to $\alpha = 0.2$ and Fig. 2(a) shows that 3DHoPD starts with a recall of ≈ 0.9 while PCA-USC (18 dim) starts with a recall of ≈ 0.65. The same observation can be made from Fig. 2(b) that 3DHoPD starts with a recall of ≈ 0.45 while PCA-USC(18 dim) starts with a recall of ≈ 0.1 at the same value of $\alpha = 0.2$. This high recall of 3DHoPD at lower α validates the claim that a list of probable matches with high recall is first generated (by transforming the 3D keypoints to a new 3D space) and then the exact match is later found with HoPD descriptor, offering state-of-the-art performance. The reasons behind HoPD and 3DHoPD offering nearly same performance is because in this dataset, the scene and the model point clouds are exactly the same, except that the scenes are corrupted with noise, which can be very well captured by the proposed 15-dim HoPD descriptor. However, in the following experiments on other datasets that cater for occlusion and background clutter, the enhancement offered by the proposed 3D keypoint transformation stands out significantly.

Performance on Random Views dataset: Fig. 3(a) and (b) show the performance of 3DHoPD, HoPD and the low dimensional state-of-the-art descriptors on the Random Views dataset with two settings, where the scenes are corrupted with Gaussian noise of $\sigma = \{0.1, 0.5\}mr$. Fig. 3(a) shows that for $\alpha > 0.925$, 3DHoPD offers both high precision and recall compared to all other descriptors. For descriptor based keypoint matching, having a high recall is preferable because RANSAC can later be used to improve the precision by removing the false positive correspondences. Moreover, $\alpha = 1$ boils down to the nearest neighbour association, which is the most preferred type of descriptor matching in practical applications. It can be seen from Fig. 3(a) that 3DHoPD offered high precision and recall at the most practical operating point, $\alpha = 1$, in real world applications. In the case with higher amount of Gaussian noise ($\sigma = 0.5mr$), as shown in Fig. 3(b), 3DHoPD offered better precision and recall for $\alpha > 0.9$, when compared to others. At a few values of α in both Fig. 3(a) and (b), PCA-3DSC has offered good performance in this dataset, however, it performed poorly on all other

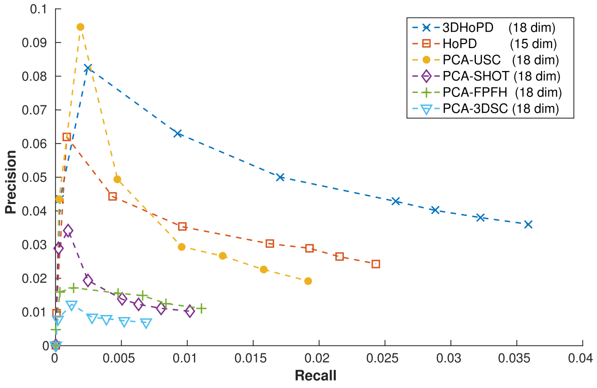


Fig. 4. Performance on Queens dataset.

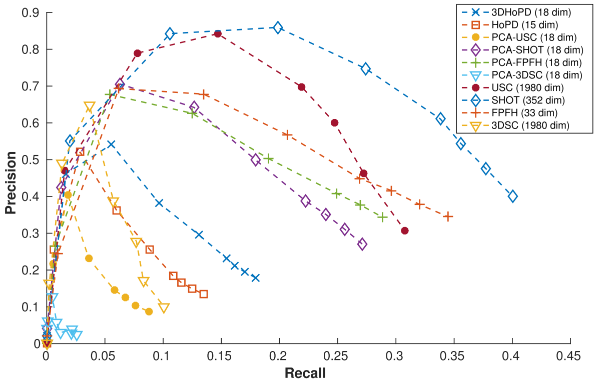


Fig. 5. Performance on Kinect dataset.

datasets. It can be seen from Fig. 3(a) and (b) that 3DHoPD offers much better performance than HoPD, which is because of the proposed 3D keypoint transformation method. As, higher recall is preferred in practical scenarios so that the descriptors can provide more number of true matches, it can be said that 3DHoPD is preferable over others because it consistently offers higher recall across different noise settings.

Performance on Queens dataset: Fig. 4 shows the performance on Queens dataset and it can be seen that 3DHoPD outperforms all the considered 3D descriptors significantly. This experiment again bolsters the fact that the proposed 3D keypoint transformation significantly enhances the performance of HoPD, as can be seen by the difference in performance of 3DHoPD and HoPD descriptors in Fig. 4. PCA-USC offers the third best performance followed by PCA-FPFH, PCA-SHOT and PCA-3DSC.

Performance on Kinect dataset: It was observed that the original Kinect dataset does not have significant rotation between the models and the scenes. Hence to accommodate and evaluate the rotational invariance of the considered 3D descriptors, we applied a random 3D transformation between each model-scene pair and then performed the experiments. We also present the performance comparison with full dimensional USC, SHOT, FPFH and 3DSC descriptors for this dataset, whereas the results for other datasets are available on the project website. It can be seen from Fig. 5 that full dimensional SHOT, USC and FPFH offer better performance than 3DHoPD. However, please note the huge dimensionality of USC (1980 dim) and SHOT (352 dim) while compared to 3DHoPD (18 dim). Next, FPFH requires nearly 275x more computational resources than 3DHoPD, as shown in Table II. While comparing 3DHoPD with 18-dim PCA equivalents, then 3DHoPD comes after PCA-SHOT and PCA-FPFH. PCA-SHOT's and PCA-FPFH's success can firstly be attributed to their design which captures the variations in surface normals and secondly, to the specific characteristic of this dataset for having less surface variations that get accurately captured by surface normals rather than spatial point distributions. However, these PCA-SHOT and PCA-FPFH offer significantly lower performance than 3DHoPD in other datasets while 3DHoPD offers most stable performance among all the considered 3D descriptors across all the datasets.

TABLE II
AVERAGE NUMBER OF KEYPOINTS AND THE COMPUTATIONAL TIME (IN SECONDS) REQUIRED FOR EXTRACTING FEATURE DESCRIPTORS AROUND THOSE KEYPOINTS ON THE BOLOGNA KINECT DATASET

No. of Keypoints	3DHoPD	USC	SHOT	3DSC	FPFH
1342	**1.12 sec**	9.48	14.52	25.17	276.12

Computational Requirements For Descriptor Extraction: We employed a CPU with an *Intel Xeon(R) CPU E5-16500@* 3.20 GHz × 12 and 16 GB RAM with *UBUNTU 14.04* operating system. The ISS keypoints were detected and 3D descriptors were extracted with exactly same settings as in previous experiments on the Kinect dataset, which has a mesh resolution of $\approx$0.001 m. Table II reports the average number of keypoints detected and the time taken to extract 3D descriptors around them. It can be seen that to extract 1342 descriptors, 3DHoPD requires just 1.12 sec (10x faster) while all others require significantly higher computational time. The average number of points in the local patch used for descriptor extraction in our experiments turned out to be 12300. The high computational requirements of FPFH descriptor can be justified from the survey letter [10], which mentions that FPFH is only efficient when the number of points are lower than 5000 and its computational requirements increase later on. Also note that for other descriptors, we did not account the time taken to create low dimensional representations (through PCA transformation matrices) and only reported the time taken to extract full dimensional descriptors. 3DHoPD has a great computational edge over other descriptors as it does not involve expensive surface normal estimation. This shows that 3DHoPD stands out from the existing 3D descriptors in terms of fast descriptor extraction, low memory footprint and stable performance across various datasets.

Memory: According to IEEE 754 floating point standard, single 18-dim 3DHoPD requires 18*4 = 72 bytes of memory.

IV. OBSERVATIONS AND DISCUSSIONS

Firstly, USC, SHOT and the proposed 3DHoPD descriptors employ the same technique to estimate the local reference frame and then transform the 3D surface patch accordingly to achieve rotational invariance.

Secondly, 3DSC and USC descriptors create huge number of partitions (1980) in the superimposed spherical grid and represent each grid with just a single value calculated from the weighted sum of points that fall in each partition. In this way, they lose out on actual surface variations along x, y, z directions by only accounting for the 'number' of points present in each partition, which can be reason for their low performance. SHOT descriptor has less number of partitions compared to USC, however, it represents each partition with a histogram of normals, which does capture low level surface variations as observed in the Kinect dataset, however, coarse partitioning of spherical grid and not accurately accounting for the number of points is the reason for its low performance on other datasets. FPFH strongly relies on normals for Darbaux frame computation and hence offers good performance on Kinect dataset but it loses out on other datasets as it does not directly and accurately account for spatial distributions and variations of the surface points along x, y, z directions.

Our previous work [25] shows that, after PCA based dimensionality reduction, about 70% of SHOT descriptors's performance is retained in the first 20 dimensions while 95% of FPFH descriptor's performance is retained in the first 20 dimensions, of their low dimensional representations. Hence, dimensionality reduction also accounts for the loss in performance of the descriptors on these datasets. Finally, the competitive performance of 3DHoPD across multiple datasets can be strongly attributed to the proposed 3D keypoint transformation to a new 3D space from which a small list of highly probable matches can be retrieved, effectively getting rid of large number of false positives and hence making the job for descriptor matching easier. Next, HoPD actually considers the surface variations along x, y, z directions and also accounts for the number of points through histogram normalization. However, histogram based binning does not account for little surface variations (which normals are good at) and hence offers reduced performance on the Kinect dataset. 3DHoPD's stable performance across datasets and its very low memory and computational requirements, compared to other descriptors, makes it a preferable choice in mobile applications.

V. Conclusion

This letter introduced a new 18-dimensional 3D descriptor, 3DHoPD, which has low memory footprint and is extremely fast to compute (10x faster). We proposed a new idea, in which, the 3D keypoints are transformed to a new 3D space, where keypoints arising from similar 3D surface patches lie close to each other. Capitalizing on this idea, for a given source keypoint, a list of probable target keypoint matches are retrieved with high recall, effectively reducing the search space by 90%. Then, histogram of point distributions (HoPD) is proposed to find an exact correspondence from the found list of probable matches. Experiments have showed that 3DHoPD offers stable and competitive performance across multiple datasets and has dramatically low computational costs, when compared to existing state-of-the-art 3D descriptors at similar dimensionality.

References

[1] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard, "3-D mapping with an RGB-D camera," *IEEE Trans. Robot.*, vol. 30, no. 1, pp. 177–187, Feb. 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 *Proc. IEEE Int. Conf. Robot. Autom.*, May 2015, pp. 4216–4223.

[3] Y. Guo, F. Sohel, M. Bennamoun, J. Wan, and M. Lu, "An accurate and robust range image registration algorithm for 3D object modeling," *IEEE Trans. Multimedia*, vol. 16, no. 5, pp. 1377–1390, Aug. 2014.

[4] A. Aldoma, F. Tombari, L. Di Stefano, and M. Vincze, "A global hypothesis verification framework for 3D object recognition in clutter," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 7, pp. 1383–1396, Jul. 2016.

[5] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, and J. Wan, "3D object recognition in cluttered scenes with local surface features: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2270–2287, Nov. 2014.

[6] M. Savelonas, I. Pratikakis, and K. Sfikas, "Partial 3D object retrieval combining local shape descriptors with global Fisher vectors," in *Proc. Eurograph. Workshop 3D Object Retrieval*, 2015, pp. 23–30.

[7] T. Faulhammer, A. Aldoma, M. Zillich, and M. Vincze, "Temporal integration of feature correspondences for enhanced recognition in cluttered and dynamic environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2015, pp. 3003–3009.

[8] R. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 1–4.

[9] S. Prakhya, L. Bingbing, Y. Rui, and W. Lin, "A closed-form estimate of 3D ICP covariance," in *Proc. 14th IAPR Int. Conf. Mach. Vision Appl.*, May 2015, pp. 526–529.

[10] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, and N. Kwok, "A comprehensive performance evaluation of 3D local feature descriptors," *Int. J. Comput. Vision*, vol. 116, pp. 66–89, 2016. [Online]. Available: http://dx.doi.org/10.1007/s11263-015-0824-y

[11] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 433–449, 1999.

[12] A. Frome, D. Huber, R. Kolluri, T. Bülow, and J. Malik, "Recognizing objects in range data using regional point descriptors," in *Proc. Eur. Conf. Comput. Vision*, 2004, pp. 224–237.

[13] F. Tombari, S. Salti, and L. Di Stefano, "Unique shape context for 3D data description," in *Proc. ACM Workshop 3D Object Retrieval*, 2010, pp. 57–62. [Online]. Available: http://doi.acm.org/10.1145/1877808.1877821

[14] Y. Guo, F. Sohel, M. Bennamoun, M. Lu, and J. Wan, "Rotational projection statistics for 3D local surface description and object recognition," *Int. J. Comput. Vision*, vol. 105, pp. 63–86, 2013.

[15] Y. Guo, F. Sohel, M. Bennamoun, J. Wan, and M. Lu, "A novel local surface feature for 3D object recognition under clutter and occlusion," *Inf. Sci.*, vol. 293, pp. 196–213, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025514009219

[16] H. Chen and B. Bhanu, "3D free-form object recognition in range images using local surface patches," *Pattern Recognit. Lett.*, vol. 28, no. 10, pp. 1252–1262, 2007.

[17] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz, "Aligning point cloud views using persistent feature histograms," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2008, pp. 3384–3391.

[18] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2009, pp. 3212–3217.

[19] S. Salti, F. Tombari, and L. Di Stefano, "SHOT: Unique signatures of histograms for surface and texture description," *Comput. Vision Image Understanding*, vol. 125, pp. 251–264, 2014.

[20] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, pp. 91–110, 2004.

[21] R. Bro, E. Acar, and T. G. Kolda, "Resolving the sign ambiguity in the singular value decomposition," *J. Chemometrics*, vol. 22, no. 2, pp. 135–140, 2008.

[22] F. Tombari, S. Salti, and L. D. Stefano, "Performance evaluation of 3D keypoint detectors," *Int. J. Comput. Vision*, vol. 102, pp. 198–220, 2013.

[23] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005.

[24] Y. Zhong, "Intrinsic shape signatures: A shape descriptor for 3D object recognition," in *Proc. IEEE 12th Int. Conf. Comput. Vision Workshops*, Sep. 2009, pp. 689–696.

[25] S. M. Prakhya, B. Liu, and W. Lin, "On creating low dimensional 3D feature descriptors with PCA," Dec. 2016. [Online]. Available: https://www.researchgate.net/publication/311927933_On_Creating_Low_Dimensional_3D_Feature_Descriptors_with_PCA