



City Research Online

City, University of London Institutional Repository

Citation: Kechagias-Stamatis, O., Aouf, N. & Chermak, L. (2017). B-HoD: A Lightweight and Fast Binary Descriptor for 3D Object Recognition and Registration. 2017 IEEE 14th International Conference on Networking, Sensing and Control (ICNSC), doi: 10.1109/ICNSC.2017.8000064 ISSN 1810-7869 doi: 10.1109/ICNSC.2017.8000064

This is the accepted version of the paper.

This version of the publication may differ from the final published version.

Permanent repository link: <https://openaccess.city.ac.uk/id/eprint/22104/>

Link to published version: <https://doi.org/10.1109/ICNSC.2017.8000064>

Copyright: City Research Online aims to make research outputs of City, University of London available to a wider audience. Copyright and Moral Rights remain with the author(s) and/or copyright holders. URLs from City Research Online may be freely distributed and linked to.

Reuse: Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge. Provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.

City Research Online:

<http://openaccess.city.ac.uk/>

publications@city.ac.uk

B-HoD: A Lightweight and Fast Binary descriptor for 3D Object Recognition and Registration

Odysseas Kechagias-Stamatis, Nabil Aouf and Lounis Chermak

Abstract—3D object recognition and registration in computer vision applications has lately drawn much attention as it is capable of superior performance compared to its 2D counterpart. Although a number of high performing solutions do exist, it is still challenging to further reduce processing time and memory requirements to meet the needs of time critical applications. In this paper we propose an extension of the 3D descriptor Histogram of Distances (HoD) into the binary domain named the Binary-HoD (B-HoD). Our binary quantization procedure along with the proposed preprocessing step reduce an order of magnitude both processing time and memory requirements compared to current state of the art 3D descriptors. Evaluation on two popular low quality datasets shows its promising performance.

Keywords—3D Binary Descriptor, 3D Object Recognition, 3D Object Registration, Local Features, Statistical Analysis.

I. INTRODUCTION

The advent of commercial low cost 3D data acquiring devices, e.g. Microsoft Kinect and Bumblebee XB3 made 3D object recognition for robotic applications an affordable option. This is important as 3D is capable of achieving high recognition performance while being less prone to external conditions such as illumination variation and pose changes [1], [2].

Most common 3D computer vision applications include, but are not restricted to: 3D object detection and/ or object recognition, scene perception, surveillance, navigation and object grasping for robotic applications [3]–[7]. These tasks require a quite accurate pose estimation of the detected and recognized object within the scene in order to properly register the template on the scene. In addition, this operation has to be executed in real time with minimum memory resources.

Although current 3D descriptors are a few and perform well [3]–[14], their computational and memory requirements may exceed the capabilities of a lightweight platform. A solution meeting those requirements can be exploiting a binary descriptor instead of a floating point as this allows a faster feature matching process along with a smaller descriptor footprint. Up-to-date 3D binary descriptors are BRAND [12], B-SHOT [13] and applying 2D binary descriptors on depth images [15]. Although

BRAND is both fast to execute and has a small memory demand, it has a feature-level fused descriptor requiring both depth and texture information. The latter, is not always affordable constraining BRAND from numerous 3D object recognition tasks. An indirect approach is suggested by Krizaj *et al.* [15]. They propose converting the 2.5D range image of the scene it into its shape index form and apply off-the-shelf 2D binary descriptors. Although their concept is promising, calculating the shape index introduces an extra processing burden that might not be affordable. Lately, Prakhya *et al.* [13] transformed the state-of-the-art floating point 3D descriptor SHOT into a binary form and suggested the B-SHOT. This remapping is achieved by forcing four consecutive values of the SHOT descriptor into a number of sum-based tests that define the binary values of the B-SHOT descriptor.

Urged from the processing and memory related advantages of a binary descriptor we propose the Binary – Histogram of Distances (B-HoD) which is an extension to the already fast to execute HoD descriptor [14]. Our solution is appealing for platforms with low hardware resource standards.

The contributions of our paper can be summarized as:

- a. Introducing a 3D binary descriptor that can be applied directly to the point cloud.
- b. A 3D descriptor that is fast to execute and has a very small memory demand making it appropriate for time-critical 3D pattern recognition and registration applications relying on low hardware resources.
- c. A combination of the processing efficient Hamming distance metric with the well performing Nearest Neighbor Distance Ratio (NNDR) matching criterion. This is unique as current binary descriptors both in the 2D and the 3D domain exploit the Hamming distance metric in combination with the inferior Nearest Neighbor Distance (NND) criterion [16].

The rest of the paper is organized in the following sections. Section II refers to our proposed 3D binary descriptor, the B-HoD. Section III compares and contrasts our approach on low quality datasets with current 3D local feature based algorithms. Finally, section IV concludes this paper.

II. B-HOD FEATURE DESCRIPTOR

Given a point cloud $P \subset \mathcal{R}^3$, each vertex can be represented as $P_i = (x_i, y_i, z_i)^T, i \in [0, K]$ where K is the total number of points. For a given set of keypoints, a spherical volume with support radius r centered on each keypoint is extracted. Then for each local area, one border

*Research supported by MBDA UK under grant HP29012014.

O. Kechagias Stamatis, N. Aouf and L. Chermak are with the Centre for Electronic Warfare Information and Cyber, Cranfield University at the UK Defence Academy, Shrivenham, SN6 8LA, UK (e-mail: {o.kechagiasstamatis, n.aouf, l.chermak}@cranfield.ac.uk)

point that is the furthest away from the corresponding keypoint is chosen as a reference point P_r . Given P_r we calculate all pairwise L2-norms to the vertices $P'_i, i \in [0, L], L \leq K$ belonging to the local area:

$$d_i = \|P_r - P'_i\|_2 \quad (1)$$

The L2-norms are in a continuous variable form and thus highly prone to even minor positioning perturbations and missing vertices. Therefore, we discretize d_i by via a static unsupervised data binning method. Considering that processing efficiency and robustness to perturbations such as subsampling are of equal importance, we discretize d_i using the equal width interval binning method [17]. This method is fast to execute and sorts the observed continuous values d_i into B equally sized bins of width δ . So, the discretized d'_i inter-distances are given by:

$$\left\{ d'_i = \left\lfloor \frac{d_i}{\delta} \right\rfloor \middle| (\forall i \in [0, \max(d_i)] \wedge \delta = \frac{\max(d_i)}{B}) \right\} \quad (2)$$

Finally, the distances d'_i are transformed into a probability mass density which is then converted into a histogram. The latter describes the local area by encrypting counters of the quantized d'_i and enhances robustness by compressing information into bins. The quantized descriptor D is defined as:

$$D = P_r \left(\left\{ s \in S : d'_i(s) = x \right\} \right) \quad (3)$$

$$\left| x \in [1, B], d'_i : S \rightarrow [1, B] \right.$$

For enhanced performance we establish a dual-layered bin size distribution scheme by defining a coarse and a fine description process of the local area. So the HoD descriptor is given by:

$$HoD = P_r \left(\left\{ s \in S : d'_i(s) = x \right\} \right) \cup P_r \left(\left\{ u \in U : d'_i(u) = y \right\} \right) \quad (4)$$

$$\left| x \in [1, 40], d'_i : S \rightarrow [1, 40], \right.$$

$$\left. y \in [1, 200], d'_i : U \rightarrow [1, 200] \right.$$

where S,U refer to the bin range of each histogram in relation to the coarse and fine description process while x,y are the bin indexes. Since HoD relies on the probability mass density it sums up to one enhancing robustness to point cloud resolution changes [6], [7].

Algorithm I: BINARY QUANTIZATION PSEUDO CODE

```

1  function Binary Transformation
   Input: Floating point number descriptor
   Output: B-HoD descriptor
2  For i=1: HoD descriptor length
3     If  $\nabla HoD > 0$ ;  $B - HoD_k = \{1\}$ 
4     else  $B - HoD_k = \{0\}$ 
       end
5  end

```

In order to reduce the total processing time and the memory footprint of the descriptor, we subsample each local area down to 1/10 the original one, apply the HoD descriptor and remap HOD to the binary domain via a Binary-Coded Decimal (BCD) scheme $HoD_{10} \xrightarrow{BCD} B - HoD_2$. Subscripts denote the numerical system each descriptor is based on which for better readability it will be omitted throughout the paper. BCD relies on the gradient of HOD with the derivative calculated pairwise between the adjacent elements. For speedup derivatives are approximated, hence:

$$B - HoD_k = \nabla HoD = \frac{\partial HoD_k}{\partial k} \approx \frac{\Delta HoD_{k,k+1}}{\delta} \quad (5)$$

where $k \in [1, B]$. Each element of the B-HoD corresponds to the attitude of the gradient as given by the pseudo code presented in Algorithm I.

It should be noted that even though by creating binary descriptors in an indirect manner i.e. remapping a floating point descriptor via a BCD scheme, induces information loss, it is a generic mean to exploit the memory and matching speedup benefits of a binary descriptor. In addition, it is worth noting that in contrast to the majority of 3D local descriptors [6]–[11], [18]–[23] both HoD and B-HoD do not require a LRF/A. Further to that, the majority of current 3D descriptors [1], [9], [10], [12], [21], [24], [25], extract a spherical volume of fixed radius r equal to a multiple of the average mesh resolution $\bar{m}\bar{r}$ of all templates under consideration. In contrast to that, HoD takes advantage of the average mesh resolution per scene (mr). Hence HoD and B-HoD have a dynamically changing support radius that is directly linked to the characteristics of each individual scene.

III. EXPERIMENTAL RESULTS

A. Experimental Setup

Given a set of model features f_i^M , a ground truth transformation and the corresponding scene features f_i^S , a scene feature is matched with all model features based on a distance metric and the NNDR criterion: if the ratio of the nearest model feature f_i^M with the second nearest f_i^M is less than a threshold τ , then the scene feature f_i^S and the model feature f_i^M are considered as a match.

Based on the established matches, we evaluate the performance of each descriptor in a qualitative manner. Evaluation relies on the estimated transformation matrix T_M between the model and the scene matched keypoints, and the ground truth transformation T_{GT} . In specific, T_M is calculated based on the Iterative Closest Point (ICP) algorithm and a point-to-point error minimization metric [26]:

$$T = \arg \min_T \left(\sum_{k=1}^K \|Rm_k + t - s_k\|_2 \right) \quad (6)$$

Table I: DESCRIPTOR PARAMETERS

Descriptor	Support radius	Descriptor Length	Implementation platform	Domain
SHOT	$40 \overline{mr}$	352	C++ (PCL)	Floating point
FPFH	$20 \overline{mr}$	33	C++ (PCL)	Floating point
3DSC	$30 \overline{mr}$	1980	C++ (PCL)	Floating point
USC	$30 \overline{mr}$	1980	C++ (PCL)	Floating point
RoPS	$40 \overline{mr}$	135	MATLAB	Floating point
HoD	$40 mr$	240	MATLAB	Floating point
B-HoD	$40 mr$	240	MATLAB	Binary
HoD (*) adopting [13]	$40 mr$	240	MATLAB	Binary

where R, t are the estimated rotation and translation matrices incorporated in the T_M , K is the number of matches and m_k and s_k are the matched model and scene keypoints in respect. The R, t combination that provides the smallest T_{error} , comprises the transformation matrix T_M . Then, considering the already known ground truth transformation between the model and the scene T_{GT} , we estimate the qualitative measure [13]:

$$T_{error} = \sqrt{\sum \sum (T_M - T_{GT})^2} \quad (6)$$

Similarly to HoD, B-HoD exploits a multi-level feature matching scheme on each description level i.e., separately for the coarse and for the fine description. The description level that provides the most matches is considered as the accepted domain which keypoint matching and hypothesis generation will rely on.

During trials, the distance metric used for the NNDR matching criterion of each descriptor presented in Table I is the one proposed by each author in the original paper. For the B-HoD descriptor, we use the Hamming distance combined with the NNDR matching criterion. In order to exploit the processing efficiency of the Hamming distance, we fully implement it in Boolean arithmetic followed by a bit count:

$$D_{\text{Hamming}} = \frac{\sum (f_i^M \oplus f_i^S)}{\sum (f_i^M \oplus f_i^S)} \quad (7)$$

It is worth noticing that B-HoD is unique in terms of combining the Hamming distance metric with the NNDR criteria (Eq. 7) since current binary descriptors exploit the less efficient Nearest Neighbor Distance metric.

Since we remap a floating point descriptor into a lower level form, information loss is induced affecting the number of correspondences achieved during the matching stage and therefore we anticipate a registration performance drop.

During trials we challenge B-HoD against current 3D pattern recognition algorithms presented in Table I. Specifically, we compare and contrast B-HoD with RoPS [6], SHOT [24], FPFH [10], 3DSC [8], USC [27], HoD [14], a binary version of HOD exploiting the quantization pipeline of [13] in combination with the subsampling of the currently proposed B-HoD descriptor. For better readability, this variant of HoD is notated as HoD (*) throughout this paper. It is important to compare B-HoD against HoD(*) as these two descriptors are identical with only exception the BCD remapping stage.

The support radius of each descriptor is independently tuned on training scenes from the Bologna dataset [24]. These scenes are non-uniformly down-sampled to $\frac{1}{2}$ their mesh resolution and Gaussian noise is added with zero mean and a standard deviation of 10% the average mesh resolution \overline{mr} [6], [7].

All trials are performed in MATLAB and in C++. Implementations in C++ are attained from the Point Cloud Library (PCL) Version 1.7.2 [28] while RoPS from MATLAB File Exchange [29]. Beyond the support radius which is tuned for best recognition performance, the rest of the parameters are fixed either to the ones originally proposed by their authors or to their PCL implementation [30]. The tuned parameter settings for all feature descriptors are presented in Table I.

Although FPFH has the smallest support radius compared to the rest of the descriptors, during tuning its performance peaked at $20 \overline{mr}$. We confirm the finding in [30] stating that FPFH performance peaks for an intermediate support radius value and beyond that its performance drops.

Since we aim at time-critical applications, we randomly select 100 keypoints from each model and extract their corresponding ones in the scene based on their a priori known ground truth transformation T_{GT} . Random keypoint selection is preferred against exploiting a keypoint detector [31] as errors of the detector can affect the descriptor [6].

B. Evaluation on the Kinect dataset

Trials are based on the Kinect dataset [24], which comprises of 51 model – scene combinations. In this paper texture information is neglected and the evaluation is based on the T_{error} metric (Eq. 6). Fig. 1 shows the T_{error} of all descriptors, with each peak representing the registration error between the 3D transformation estimated from the keypoint matches and the ground transformation. In specific, Fig. 1 shows that B-HoD, HoD and RoPS present the smallest registration error.

HoD(*) and FPFH are next to follow with a number of spikes of high T_{error} levels. It is worth noting that B-HoD has a smaller T_{error} compared to the HoD(*) revealing that our proposed BCD remapping is more efficient compared to the proposed scheme in [13]. Less accurate are SHOT, 3DSC and USC which attain the highest registration errors.

Focusing on the high performing ones (B-HoD, HoD and RoPS) we observe from Fig. 2 that B-HoD has almost the same performance as HoD and achieves constantly a lower T_{error} compared to RoPS. A direct comparison between B-HoD and HoD reveals that the performance loss due to the

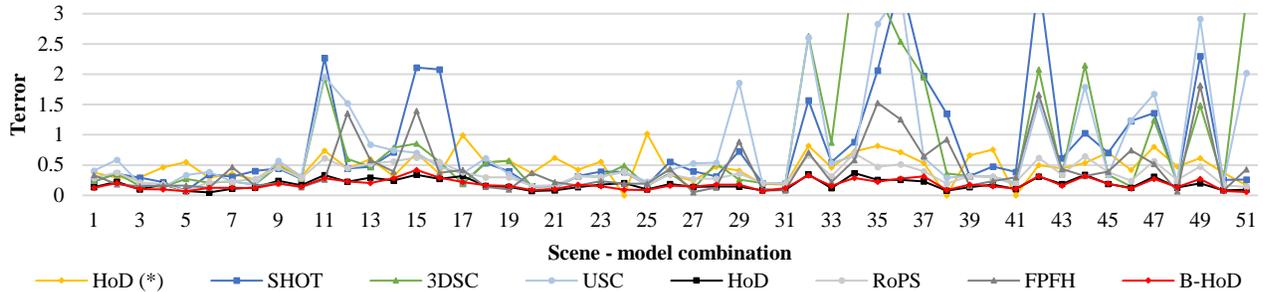


Fig. 1: Qualitative performance evaluation based on the T_{error} metric (best seen in color). Peak values exceeding a T_{error} value of 3 are truncated for better readability.

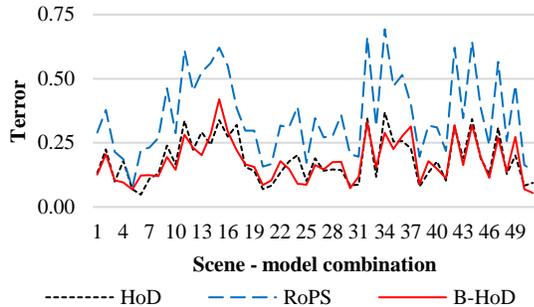


Fig. 2: Qualitative performance evaluation based on the T_{error} metric focusing on the top 3 performing ones.

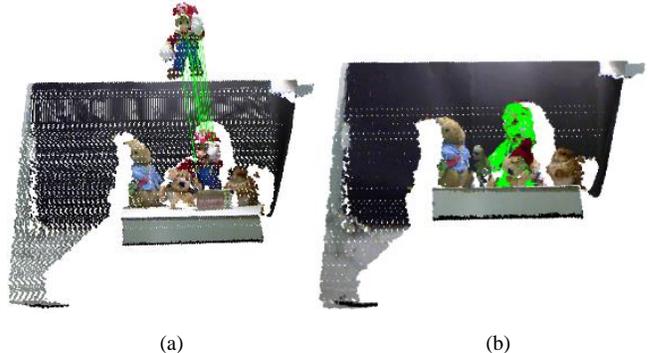


Fig. 3: Example of B-HoD on the Kinect dataset (a) Green lines indicate correct matches (b) Model point cloud (in green) is registered on the scene point cloud. Scene and template are presented with texture for readability purposes.

area subsampling and the BCD remapping is minor, showing that B-HoD is quite promising. A recognition and registration example of the B-HoD on the Kinect dataset is presented in Fig. 3.

C. Processing Efficiency

Our main interest is 3D object recognition and registration for time-critical applications based on low quality data. Thus, we investigate the processing efficiency of B-HoD against the descriptors presented in Table I.

Even though all HoD variants include real-time point resolution estimation and template – scene keypoint description, neglecting the LRF estimation reduces greatly the processing time. It is expected that B-HoD will further reduce processing time due to two additional factors. First, the local area is subsampled and second, the feature matching problem is based on the efficient Hamming distance. Indeed, B-HoD is the most efficient 3D descriptor among the ones evaluated with a large margin. Specifically, a direct B-HoD – HoD comparison reveals that B-HoD is more than 7.5 times faster compared to HoD with a processing time of 0.85ms/keypoint. It is worth noting that all HoD variants and RoPS are MATLAB implemented while the rest are in C++ providing to the former a processing setback purely due to the implementation platform. Even in that case, B-HoD is more than x40 faster compared to SHOT which is the fastest one implemented in C++. Fig. 4 presents the processing timings while for completeness, we further analyze the execution time of

each sub-process of the B-HoD and HoD descriptor. From Fig. 5 we perceive that the vast processing speedup is obtained via the local area subsampling that is incorporated within the B-HoD. In addition, the NNDR Hamming based matching scheme reduces matching time down to 25% compared to the original floating point NNDR matching.

D. Memory Consumption

Another important factor is the memory required to store the descriptor. Memory demand is highly related with the descriptor’s size and domain; therefore, we examine the memory demand in Kilobytes (Kb) per descriptor.

As expected B-HoD and HoD (*) have the smallest memory footprints of only 0.24 Kb/keypoint due to their binary nature. Although not binary, but purely due to the small descriptor size, FPFH closely follows with 0.26Kb/keypoint. The minor memory requirement of HoD is highly appealing especially for memory constrained applications. Having a descriptor with a small memory requirement allows increasing the number of templates and thus the efficiency of the application. Detailed memory demands per descriptor are presented in Fig. 6.

E. Evaluation on the SpaceTime stereo dataset

We further evaluate the B-HoD descriptor on the SpaceTime dataset [24] which consists of 24 scene – model combinations. Trials consider the parameter setup as presented in Table I and texture information is neglected. Fig. 7 shows an object recognition and registration example where it can be clearly seen that B-HoD affords a decent

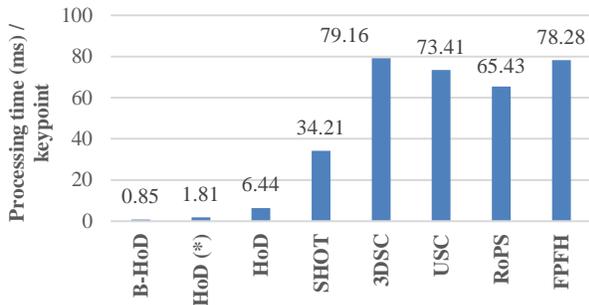


Fig. 4: Processing efficiency of the proposed and current descriptors.

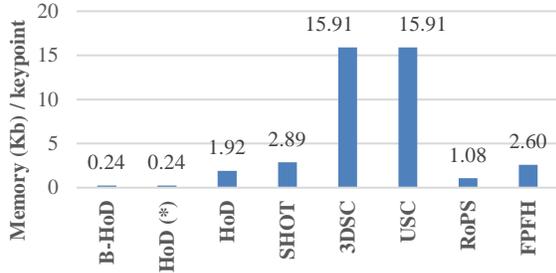


Fig. 6: Memory consumption of the proposed and state-of-the-art descriptors.

keypoint matching capability that leads to a low registration error.

Fig. 8 presents the T_{error} per descriptor per scene – model combinations. A conclusion that can be drawn is that all descriptors have an inferior T_{error} metric compared to their corresponding performance on the Kinect dataset, due to the low quality data of the SpaceTime dataset. B-HoD, RoPS and USC are the ones performing best as they offer the smallest registration error, with the latter having a few T_{error} spikes. Next to follow are HoD, SHOT and FPFH, while less accurate are HoD(*) and 3DSC. Focusing on the high performing ones i.e. B-HoD, USC and RoPS we observe that B-HoD has the smallest T_{error} with some minor fluctuations. This is important because the next two best performing ones have a very large processing burden and memory requirement compared to the proposed B-HoD. A direct comparison among the B-HoD and the HoD reveals that the B-HoD has an enhanced performance in the SpaceTime dataset. This can be explained by the fact that SpaceTime is a low quality dataset and therefore quantizing the histogram of distances into a compact form can compensate for smaller T_{error} values. Focusing on the high performing descriptors (B-HoD, USC and RoPS) we observe from Fig. 9 that B-HoD achieves the lowest T_{error} on almost every scene.

A direct performance comparison between the two datasets reveals the performance hierarchy remains almost the same. A constantly small overall T_{error} is afforded by the B-HoD and the RoPS. It should be noted though that B-HoD is more than 75 times faster and its memory footprint is 4.5 times smaller compared to RoPS.

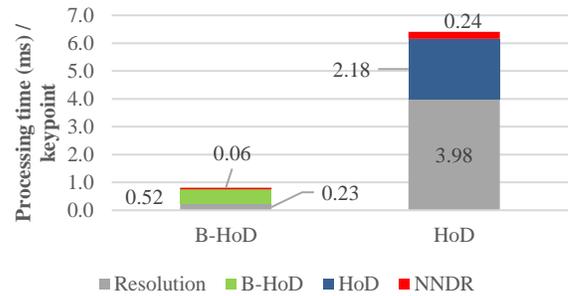


Fig. 5: Direct processing time comparison of the B-HoD and HoD per sub-process.

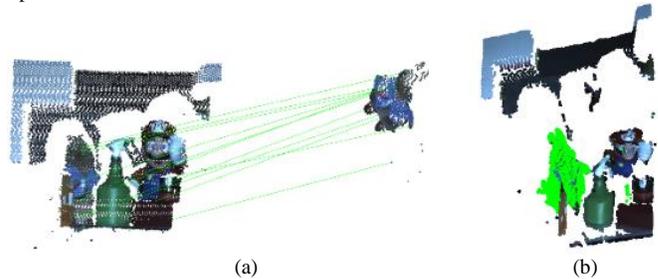


Fig. 7: B-HoD applied on the SpaceTime stereo dataset (a) Green lines indicate correct matches (b) Model point cloud (in green) is registered on the scene. Scene and template are presented with texture for readability purposes.

IV. CONCLUSION

We present a binary 3D descriptor, named the Binary Histogram of Distances (B-HoD), which is computationally efficient and requires low memory resources. We challenge B-HoD with a number of local 3D descriptors, including state-of-the-art ones, on two popular low resolution datasets, the Kinect and the SpaceTime stereo. We conclude that B-HoD maintains the registration error to a low level via an efficient BCD remapping scheme that exploits the NNDR match metric in combination with the Hamming distance. Specifically, B-HoD achieves a speed up and reduces the memory demand by an order of magnitude compared to the already fast floating point HoD descriptor.

Based on the low registration error and speedup achieved as well as on the minor memory requirement, B-HoD can be considered as an appealing solution for time-critical 3D computer vision based applications.

REFERENCES

- [1] A. S. Mian, M. Bennamoun, and R. Owens, “Three-Dimensional Model-Based Object Recognition and Segmentation in Cluttered Scenes,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 10, pp. 1584–1601, Oct. 2006.
- [2] Y. Lei, H. Lai, and X. Jiang, “3D face recognition by SURF operator based on depth image,” in *Proceedings - 2010 3rd IEEE International Conference on Computer Science and Information Technology, ICCSIT 2010*, 2010, vol. 9, pp. 240–244.
- [3] B. Steder, R. B. Rusu, K. Konolige, and W. Burgard, “Point feature extraction on 3D range scans taking into account object boundaries,” in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 2601–2608.
- [4] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, “Fast 3D recognition and pose using the Viewpoint Feature Histogram,” in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 2155–2162.

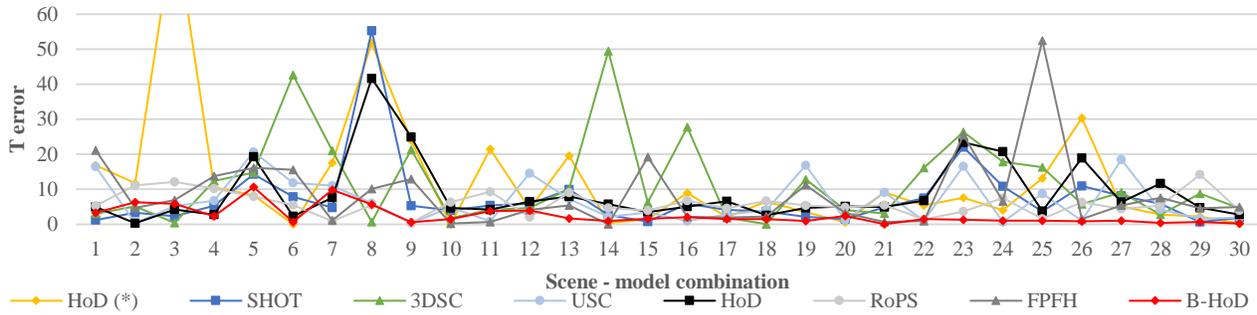


Fig. 9: Qualitative performance evaluation on the SpaceTime dataset based on the Terror metric (best seen in color). A peak value of the HoD(*) has been truncated for better readability.

[5] O. Kechagias-Stamatis and N. Aouf, "Fast 3D object matching with Projection Density Energy," in *2015 23rd Mediterranean Conference on Control and Automation (MED)*, 2015, pp. 752–758.

[6] 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. Vis.*, vol. 105, no. 1, pp. 63–86, Oct. 2013.

[7] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," *Comput. Vision–ECCV 2010*, pp. 356–369, 2010.

[8] A. Frome, D. Huber, R. Kolluri, T. Bülow, and J. Malik, "Recognizing Objects in Range Data Using Regional Point Descriptors," in *ECCV*, vol. 3023, 2004, pp. 224–237.

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

[10] R. B. Rusu, N. Blodow, and M. Beetz, "Fast Point Feature Histograms (FPFH) for 3D registration," in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 3212–3217.

[11] J. Sun, M. Ovsjanikov, and L. Guibas, "A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion," *Comput. Graph. Forum*, vol. 28, no. 5, pp. 1383–1392, Jul. 2009.

[12] E. R. Nascimento, G. L. Oliveira, M. F. M. Campos, A. W. Vieira, and W. R. Schwartz, "BRAND: A robust appearance and depth descriptor for RGB-D images," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 1720–1726.

[13] S. M. Prakhya, B. Liu, and W. Lin, "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)*, 2015, pp. 1929–1934.

[14] O. Kechagias-stamatis and N. Aouf, "Histogram of Distances for Local Surface Description," in *Robotics and Automation (ICRA), IEEE international Conference*, 2016, pp. 2487–2493.

[15] V. Struc, "A Feasibility Study on the Use of Binary Keypoint Descriptors for 3D Face Recognition," in *In Mexican Conference on Pattern Recognition*, 2014, pp. 142–151.

[16] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *2003 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition, 2003. Proceedings.*, vol. 2, pp. II–257–II–263.

[17] R. B. Rusu, A. Holzbach, M. Beetz, and G. Bradski, "Detecting and segmenting objects for mobile manipulation," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference*, 2009, pp. 47–54.

[18] Yulan Guo, M. Bennamoun, F. Sohel, Min Lu, and Jianwei 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.

[19] Y. Zhong, "Intrinsic shape signatures: A shape descriptor for 3D object recognition," in *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, 2009, pp. 689–696.

[20] A. S. Mian, M. Bennamoun, and R. A. Owens, "A Novel Representation and Feature Matching Algorithm for Automatic Pairwise Registration of Range Images," *Int. J. Comput. Vis.*, vol. 66, no. 1, pp. 19–40, Jan. 2006.

[21] Y. Guo, F. Sohel, M. Bennamoun, M. Lu, and J. Wan, "TriSI: A Distinctive Local Surface Descriptor for 3D Modeling and Object Recognition," in *8th International Conference on Computer*

Graphics Theory and Applications, 2013.

[22] F. Tombari, S. Salti, and L. Di Stefano, "A combined texture-shape descriptor for enhanced 3D feature matching," in *2011 18th IEEE International Conference on Image Processing*, 2011, pp. 809–812.

[23] J. Novatnack and K. Nishino, "Scale-Dependent/Invariant Local 3D Shape Descriptors for Fully Automatic Registration of Multiple Sets of Range Images," in *ECCV '08 Proceedings of the 10th European Conference on Computer Vision: Part III*, 2008, vol. 5304, pp. 440–453.

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

[25] Yulan Guo, F. a. Sohel, M. Bennamoun, Jianwei Wan, and Min Lu, "RoPS: A local feature descriptor for 3D rigid objects based on rotational projection statistics," in *2013 1st International Conference on Communications, Signal Processing, and their Applications (ICCSIPA)*, 2013, pp. 1–6.

[26] F. Pomerleau, F. Colas, and R. Siegwart, "A Review of Point Cloud Registration Algorithms for Mobile Robotics," *Found. Trends Robot.*, vol. 4, no. 1, pp. 1–104, 2015.

[27] F. Tombari, S. Salti, and L. Di Stefano, "Unique shape context for 3d data description," in *Proceedings of the ACM workshop on 3D object retrieval - 3DOR '10*, 2010, p. 57.

[28] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 1–4.

[29] "MATLAB Central FileExchange." [Online]. Available: <http://www.mathworks.com/matlabcentral/fileexchange/>. [Accessed: 20-Apr-2014].

[30] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, and N. M. Kwok, "A Comprehensive Performance Evaluation of 3D Local Feature Descriptors," *Int. J. Comput. Vis.*, vol. 116, no. 1, pp. 66–89, Jan. 2016.

[31] F. Tombari, S. Salti, and L. Di Stefano, "Performance evaluation of 3D keypoint detectors," *Int. J. Comput. Vis.*, vol. 102, no. 1–3, pp. 198–220, 2013.

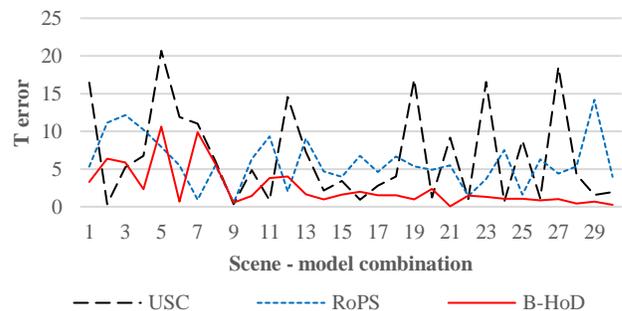


Fig. 8: Terror metric on the SpaceTime dataset focusing on the top 3 performing descriptors.