Pattern Recognition Letters 83 (2016) 42–48

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Detecting keypoint sets on 3D point clouds via Histogram of Normal Orientations☆

Sai Manoj Prakhya[a], Bingbing Liu[b,*], Weisi Lin[a]

[a] School of Computer Engineering, Nanyang Technological University, Singapore
[b] Institute for Infocomm Research, A*STAR, Singapore

ARTICLE INFO

Article history:
Available online 16 June 2016

Keywords:
Keypoint detection
3D point cloud
Histogram of Normal Orientation
Keypoint group
Feature

ABSTRACT

We propose a method to detect keypoint sets on 3D point clouds. Contrary to the existing norm that each and every keypoint should be distinctive, most of the keypoints detected by the proposed method lie in groups (from now, we refer to 'groups' as 'sets') and these groups (sets) of keypoints are distinctive. It is reliable to have sets of keypoints at high curvature and more informative areas rather than having a single keypoint that might sometimes arise due to noise. The proposed algorithm has two well-defined steps for keypoint sets detection. Firstly, Histogram of Normal Orientations (HoNO), which is calculated at every point in the point cloud is employed to avoid planar regions and successfully detect salient regions. Secondly, the keypoint sets are detected from the salient regions by evaluating the properties of both the HoNO and the neighborhood covariance matrix. Through extensive experiments on publicly available benchmark datasets, it is shown that the detected keypoint sets offer better repeatability than those by the existing ones.

1. Introduction

3D keypoint detection is heavily employed in various applications such as 3D object recognition [7,14], Simultaneous Localization and Mapping [5], Sparse Depth Odometry [12], 3D shape retrieval and point cloud registration [11,23]. A keypoint detector reduces an input point cloud to fewer number of keypoints, which helps in improving the accuracy and reducing the computational requirements of the target application.

Tombari et al. [21] have presented an extensive performance evaluation of various existing 3D keypoint detectors. Existing 3D keypoint detectors can be classified either as fixed or adaptive scale keypoint detectors. A fixed scale keypoint detector works on a constant scale whereas an adaptive scale keypoint detector generates multiple scales of the input point cloud and detects keypoints on various scales. Adaptive scale 3D keypoint detectors are motivated by the success of the SIFT keypoint detector [8] in the 2D image domain. It is to be noted that generating a scale space representation of a 3D point cloud deforms the 'structure' of the input point cloud. A fixed scale keypoint detector operates on a local neighborhood of radius ϵ at a constant scale. It can also be seen that, varying the parameter ϵ would enable a fixed keypoint detector to detect keypoints at various scales. In this paper, we propose a fixed scale keypoint detector that leverages on local neighborhood properties to detect repeatable keypoints. Please note that we work on unordered 3D point clouds and do not assume the presence of RGB or mesh connectivity information.

As mentioned by Tombari et al. [21], a fixed-scale keypoint detector generally comprises of two steps. First one is a pruning step and the second one is a non-maxima suppression (NMS) step. In the pruning step, most of the ineffective information is discarded based on some criteria and the informative points are passed to the next stage. In the NMS step, yet another test is performed in a radius of R_{nms} over the informative points to detect the final keypoints that are distinctive and are atleast a distance of R_{nms} away from each other.

Local Surface Patches (LSP), as developed by Chen and Bhanu [3] employed Shape Index metric (SI) [4] to prune non-informative points and find the salient points. NMS is performed on the found salient points and only those salient points are labeled as keypoints, which attain either a local maxima or minima in terms of SI. Zhong [25] developed ISS keypoint detector that uses the two successive eigenvalues of the neighborhood covariance matrix for pruning while considering the smallest eigenvalue for NMS. Mian et al. [9] presented KeyPoint Quality (KPQ) detector that employed a relaxed pruning step to detect more salient points when compared to ISS keypoint detector and used curvatures in the NMS

☆ This paper has been recommended for acceptance by Chennai Guest Editor.

* Corresponding author. Tel.: +65 6408 2695.
E-mail addresses: bliu@i2r.a-star.edu.sg, liubb2004@gmail.com (B. Liu).

http://dx.doi.org/10.1016/j.patrec.2016.06.002
0167-8655/

step to detect final keypoints. Fiolka et al. [6] developed SURE keypoint detector that works by calculating the entropy of surface normal histograms and their variance in a local neighborhood to detect keypoints. NARF keypoint detector as proposed by Steder et al. [19] generates a range image and finds keypoints at stable and curved regions by moving the keypoints away from the object boundaries present in the scene. SDTP keypoint detector [24] calculates interest values by employing signed distances on a tangent plane and a surface curvature based NMS to detect keypoints. While the research is heading towards developing more robust and repeatable keypoints, interestingly, [1,2] have showed that by employing game-theoretic matching of simple surface 3D keypoints, a robust surface registration can be achieved even without any initial estimate, which is pre-requisite for ICP ([15,23]) algorithm.

The two important traits that are expected from a good 3D keypoint detector are:

- The detected keypoints should not lie on planar regions because feature descriptor based keypoint matching would become highly ambiguous.
- The pruning and the NMS steps should be robust enough to not detect noise and glitches as a keypoint.

The proposed keypoint detector differs from the existing literature in terms of the criteria that are employed in the two steps, namely salient region detection and pruning. The proposed salient region detection step efficiently removes planar or non-informative regions from the input point cloud and the pruning step condenses the salient regions to sets of keypoints. Moreover, it is better to have more keypoints at high curvature or informative areas rather than having distinctive keypoints that may sometimes arise due to noise. Hence, contrary to the existing norm that each and every keypoint should be distinctive, we propose to find keypoint sets that are distinctive, depending on the target scene. We provide robust criterion for these steps and show that the detected keypoint sets are more repeatable than the relevant existing ones on 3D keypoint detection benchmark datasets.

The detected keypoints essentially represent informative 'places' and 'points' present in the scene. This also opens a new research direction that aims at establishing accurate 'place' matches by developing feature descriptors that employ information from neighboring keypoint sets. In the next section, we describe the proposed method for 3D keypoint sets detection.

2. The proposed keypoint detector

There are two main steps in the proposed keypoint detector. Firstly, salient region detection that removes planar regions and retains only the curved/informative regions in the input point cloud. Secondly, a pruning step that reduces the salient regions to final keypoint sets. In these two steps, we employ a technique called HoNO (Histogram of Normal Orientations), which we will first discuss, and later explain the two steps in detail. Though HoNO has been previously used for creating 3D feature descriptors [17,20,22], our novelty lies in employing them for 3D keypoint detection by estimating their spread or peakedness from the kurtosis of the histogram.

2.1. HoNO: Histogram of Normal Orientations

At every point $p_i \in P$ (input point cloud), a normal N_i is calculated by approximating it to the smallest eigenvector of the neighborhood covariance matrix C_i calculated from the points p_j, where $||p_j - p_i|| < R_{normal}$, and R_{normal} is the radius used for normal computation. After computing the normals $N_i \forall p_i$, a histogram H_i that represents the orientations of neighborhood normals N_j with respect to N_i is calculated for every point p_i, where N_j are the normals that belong to the neighborhood points p_j such that $||p_j - p_i|| < R_{HoNO}$ while R_{HoNO} is the radius used for HoNO computation. The orientation of a specific normal N_q with respect to N_i, where $N_q \in N_j$, represented by an angle θ_{deg} is calculated as shown in Eq. (1).

$$\theta_{deg} = \left(\arctan \frac{||N_i \times N_q||}{N_i \cdot N_q} \right) * \frac{180}{\pi} \tag{1}$$

where $\times$ represents the cross product and $\cdot$ represents the dot product between two vectors.

The angle θ_{deg} lies between 0 to 180 as $N_i \cdot N_q$ is always non-negative. The angle θ_{deg} is used to fill the histogram H_i of 18 bins with each bin size of 10 degrees. After binning the angles of all neighborhood normals, the histogram is normalized to make it independent of neighborhood density. Please note that each point $p_i \in P$ will have an histogram H_i created from its neighborhood normal orientations. Hence we call the histogram H_i as HoNO (Histogram of Normal Orientations). Each bin of H_i has a width of 10 degrees and gives a count of neighborhood normals that fall into a specific bin. In short, this HoNO, or H_i at every point p_i represents the diversity of normals that are present in the neighborhood of p_i. It can be easily seen that a point with a planar neighborhood will have a histogram whose bin values will be high in the first bin and a zero value in rest of the bins. In the same way, a highly curved area will have most of the bins occupied in its histogram as it will have diversely oriented normals in its neighborhood. To quantitatively measure the spread or peakedness of HoNO, which represents the neighborhood normals orientation, we employ the kurtosis of an histogram as shown in Eq. (2)[1].

$$Kurt(H) = \frac{\sum_{k=1}^{N} (H_k - \bar{H})^4}{NS^4} - 3 \tag{2}$$

$$S = \sqrt{\frac{\sum_{k=1}^{N} (H_k - \bar{H})^2}{N}} \tag{3}$$

where H is the histogram under consideration, N = 18 in our case as the considered histogram has 18 bins, H_k represents the count in the k^{th} bin of H, $\bar{H}$ represents the mean of all the bins in H and S represents the standard deviation of histogram bins, which is calculated as shown in Eq. (3).

2.2. Keypoint detection

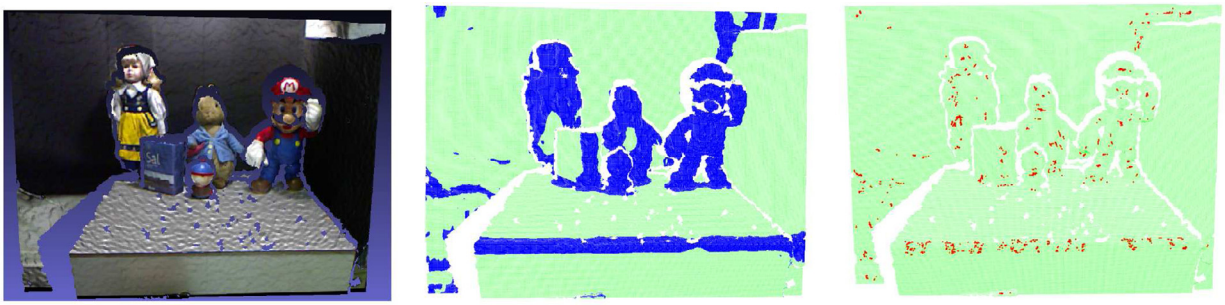
In this section, we describe the salient region detection step and the pruning step that employ HoNO and some neighborhood properties, to detect keypoints on the input point cloud.

Lets represent the input point cloud as P. We first compute normals and calculate HoNO's for each point $p_i \in P$. As explained in Section 2.2.1, salient regions in the input point cloud P are detected and those salient points that lie closer to boundaries are removed. Finally, as explained in Section 2.2.2, a pruning step is applied on the detected salient regions to find the keypoint sets. These two steps, namely salient region detection and pruning are illustrated in Fig. 1 on a scene `scene_005` from the Kinect dataset. The detected salient regions are visualized in Fig. 1(b) while the final keypoint sets after pruning are shown in Fig. 1(c).

2.2.1. Salient region detection

We first compute normals $N_i \forall p_i$ and calculate a HoNO H_i for each point $p_i \in P$ (input point cloud). The input point cloud is scanned to remove planar or less curved regions as it is unreliable to extract keypoints in those areas. This is performed by thresholding the kurtosis of HoNO at every point. The points in

[1] We use the 'excess kurtosis' formula of an histogram that reduces the kurtosis of a standard normal distribution to zero.



(a) Input Point Cloud (b) Salient Region Detection (c) Final Keypoint Sets after Pruning

Fig. 1. Keypoint sets detected by the proposed approach on `scene_005` from the Kinect dataset, which is proposed by Tombari et al. [21].

the point cloud whose histograms have a value of kurtosis higher than a threshold Th_K are not passed to the next stages as a peaked histogram has a higher value of kurtosis compared to the one that is more flat. In our case, we are interested in the regions that are more curved in nature, hence we are looking for histograms that have a wider spread, or in other words, less-peaked histogram. A less-peaked histogram is constructed from a neighborhood that has more diversely oriented surface normals, which represents a curved surface. The crafted technique of using the kurtosis of HoNO, fares well with the noisy data provided by Kinect-style depth cameras and it removes noisy planes and unreliable surfaces on the objects.

The salient regions that lie close to the boundaries are also removed as the data provided by various sensors is extremely noisy in boundary regions. Please refer to PCL library [16] for more information on boundary extraction. In particular, there is a parameter called boundary radius R_B that has to be set for boundary extraction. We use the same radius R_B to check if any point from the salient region lies close to a boundary. If yes, then that point is discarded from the salient region.

2.2.2. Pruning step

Let d_i represent the set of points from salient regions, which are passed from the previous step. For every point d_i, we check for two conditions to label it as a keypoint or not. The first condition is that in the neighborhood of radius R_d around d_i, d_i should have the lowest value of the kurtosis of HoNO. It means that at the current point d_i, the change in normals reaches a local maximum. Next, in the second condition, we check if d_i has the strongest normal, i.e., if d_i has the largest value for the smallest eigenvalue of the neighborhood covariance matrix C_{di}, compared to all the points that lie in radius R_d. If any of these two conditions hold, when compared to all the neighborhood points, then d_i is labeled as a keypoint.

One way of looking at our proposed methodology is that we employed a robust criterion for saliency detection and used a relaxed non-maxima suppression by adding an extra condition. To not confuse the readers with the taxonomy of the pruning step and the saliency based NMS, as used in [21] for fixed-scale keypoint detectors, we have named our final step as pruning step. But it can also be viewed as a relaxed NMS step as well. The reason for naming it as pruning step is that the output of this step is not always a set of distinctive keypoints and according to [21], NMS step is aimed at producing distinctive keypoints.

Pseudocode of the proposed method is presented in Algorithm 1 to aid readers in understanding and reproducing the proposed keypoint detection method. In Algorithm 1, E_3 represents the smallest eigenvalue, g_i represents the nearest neighbors of d_i, C_{di} represents the covariance matrix of the points d_i and C_{gi} represents the covariance matrix of the points g_i.

Algorithm 1 Pseudocode for keypoint detection.

Input : Point Cloud P
procedure NORMAL ESTIMATION
  for every point $p_i \in P$ **do**
    Calculate Neigh. Covariance Matrix C_i
    Normal Ni = smallest eigenvector of C_i
  Return *Normals*
procedure HONO ESTIMATION
  for every point $p_i \in P$ **do**
    get Normals N_j of Nearest Neighbors
    Create Empty HoNO for p_i
    for every Normal $N_q \in N_j$ **do**
      Calculate θ_{deg} between N_i and N_q
      bin θ_{deg} into HoNO of p_i
    Return HoNO of p_i
  return
procedure SALIENT REGION DETECTOR
  for every point $p_i \in P$ **do**
    if Kurt(HoNO of p_i) $< Th_K$ **then**
      Salient Regions $\leftarrow p_i$
  Return Salient Regions
procedure BOUNDARY REMOVAL
  for every point $p_i \in$ Salient Regions **do**
    if $||p_i -$ nearest boundary$|| < R_B$ **then**
      Remove p_i from Salient Regions
  Return Salient Regions
procedure PRUNING
  for every point $d_i \in$ Salient Regions **do**
    get Nearest Neighbors g_i
    set TempVar = 1
    for every point g_i **do**
      if Kurt($HoNO_{di}$) <Kurt($HoNO_{gi}$) **then**
        continue
      if E_3 of $C_{di} > E_3$ of C_{gi} **then**
        continue
      elseTempVar == 0
    end
    if TempVar == 1 **then**
      Keypoints $\leftarrow d_i$
  Return Keypoints

3. Experimental evaluation

The source code of the proposed method is available at https://sites.google.com/site/honokeypointsets/.

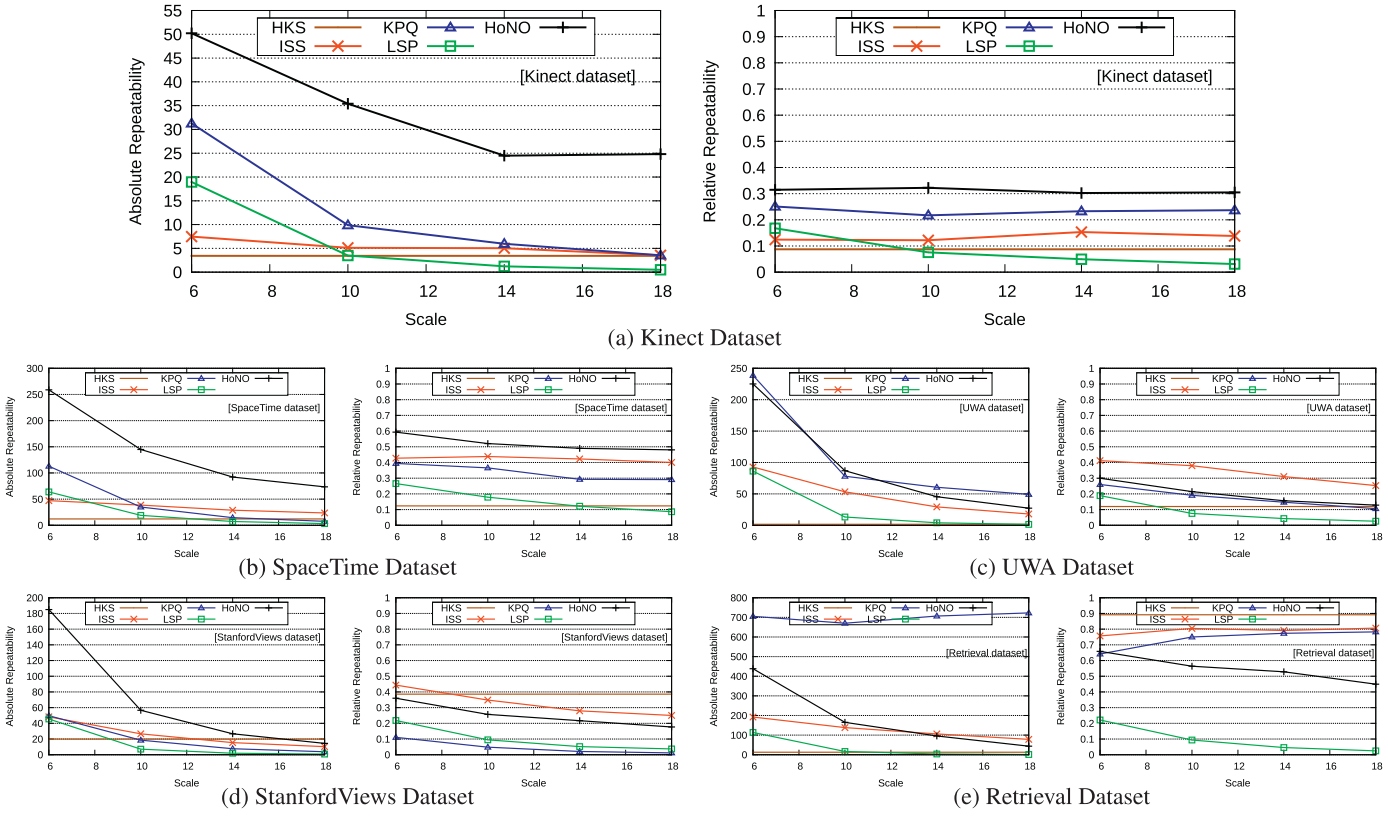


Fig. 2. Absolute and relative repeatability tests on Kinect, SpaceTime, UWA, StanfordViews and Retrieval datasets.

3.1. Evaluation metric

An important characteristic of a keypoint detector is its repeatability, which evaluates if the keypoint detector is able to find the keypoints at the same locations on an object from different views, in the presence of background, partial occlusion, clutter, etc. To capture the repeatability of a keypoint detector, the following tests as mentioned in [10,18,21] are performed. Lets consider a scene and an object model with known ground truth transformation between them. A keypoint detector under consideration detects keypoints on both scene and model. Absolute repeatability[2] calculates the number of true keypoint matches between a scene and a model. Relative repeatability is obtained as the ratio of absolute repeatability and the number of keypoints detected on the model that are not occluded in the scene.

3.2. Datasets

To evaluate the repeatability of the proposed keypoint detector and to compare it with other keypoint detectors, we use the datasets provided by Tombari et al. [21]. They provide five datasets, namely, Kinect, SpaceTime, UWA, StanfordViews and Retrieval and also the scripts[3] to evaluate and compare a new keypoint detector with others on these datasets. Kinect and SpaceTime datasets have planar backgrounds present in the scene whereas other datasets do not have any background. Retrieval and StanfordViews are synthetic datasets and we have used the datasets with Gaussian noise of 0.1x mesh resolution for evaluation purposes.

[2] A small note on calculating absolute and relative repeatability http://pointclouds.org/blog/gsoc12/gballin/repeatability.php One can also look at [21] for explanation.

[3] All the datasets and evaluation scripts are available at http://vision.deis.unibo.it/keypoints3d/?page_id=2.

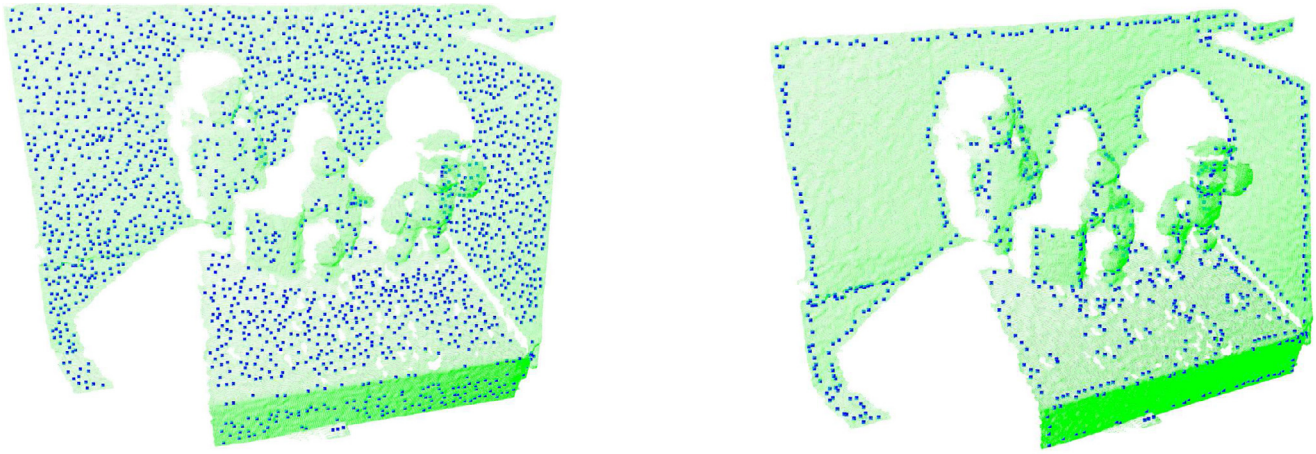
3.3. Parameters

All the radius parameters, namely, R_{normal}, R_{HoNO} and R_d are set to the same value as the scale used for experimentation. For boundary extraction, we approximately used 4x mesh resolution. Finally, Th_K, the threshold to detect salient regions is set to 6, 13, 8, 12 and 12 for Kinect, UWA, SpaceTime, Retrieval and StanfordViews datasets respectively. These values were chosen so as to detect similar number of keypoints as detected by other keypoint detectors in the comparative evaluation performed by Tombari et al. [21].

3.4. Repeatability results

Absolute and relative repeatability of the proposed keypoint detector, HoNO, in comparison with the existing ones, namely, ISS, LSP, KPQ and HKS keypoint detectors on these five datasets is shown in Fig. 2.

- It can be seen from Fig. 2(a) and (b) that HoNO outperforms other keypoint detectors on both Kinect and SpaceTime datasets.
- On UWA dataset, HoNO offers similar performance as KPQ and second-best performance for relative repeatability after ISS, as can be seen from Fig. 2(c).
- Fig. 2(d) and (e) show that HoNO keypoints offer second best relative repeatability on StanfordViews and third best on Retrieval datasets.
- ISS and KPQ keypoint detectors may seem to perform well on UWA, StanfordViews and Retrieval datasets. But they have a serious drawback of detecting keypoints on planar regions as explained below.



(a) ISS Keypoints ($Th_{21} = Th_{32} = 0.975$) (b) ISS Keypoints ($Th_{21} = Th_{32} = 0.75$)

Fig. 3. ISS keypoints detected on `scene_005` (from Fig. 1) with two different settings of Th_{32} and Th_{21} parameters.

3.4.1. Drawback of ISS and KPQ keypoint detectors

ISS keypoint detector calculates the eigenvalues E_1, E_2 and E_3 (in decreasing order) of the neighborhood covariance matrix C for every point $p_i \in P$, where P is the input point cloud. Then it checks if $\frac{E_2}{E_1} < Th_{21}$ and $\frac{E_3}{E_2} < Th_{32}$ for pruning and the NMS is performed based on the smallest eigenvalue E_3. It can be noticed that Th_{21} and Th_{32} are the parameters that influence the keypoint positions. In Fig. 3, we illustrate the ISS keypoints detected with two different parameter settings on the scene `scene_005` (shown in Fig. 1) from Kinect dataset.

ISS keypoints detected on `scene_005` with $Th_{21} = Th_{32} = 0.975$, are shown in the Fig. 3(a). It can be noticed that the keypoints are evenly spread all over the scene and even lie on the planar background. This is because, the used parameters for comparative evaluation in [21], $Th_{21} = Th_{32} = 0.975$, do not prune planar/less informative areas and the final keypoints are detected by NMS step alone. ISS keypoint detector with $Th_{21} = Th_{32} = 0.975$ parameter setting, labels the considered point as a keypoint if it has the strongest normal in a radius of R_{nms}, which might sometimes arise due to noise. It is expected that good keypoint detector should not detect keypoints that lie on planar regions/neighborhoods as the feature descriptor based keypoint matching would become highly ambiguous.

We experimented by decreasing the value of Th_{21} and Th_{32} parameters of ISS keypoint detector and it was observed that with proper parameters, planar regions can be avoided and keypoints lie in more informative areas. ISS keypoints detected on scene `scene_005` with $Th_{21} = Th_{32} = 0.75$ are shown in Fig. 3(b). It can be seen that the detected keypoints do not lie on planar regions as they are discarded in the pruning step.

We evaluated the ISS keypoint detector on UWA dataset by slowly decreasing the thresholds $Th_{21} = Th_{32} = [1, 0.9, 0.8, 0.7]$ and the relative repeatability in comparison with HoNO keypoint detector is shown in Fig. 4. Setting the thresholds to 1 means that there is no pruning at all and the keypoints are detected by NMS step alone. It can be noticed from Fig. 4 that there is significant decrease when Th_{32} and Th_{21} are changed from 1 to 0.9 and HoNO keypoint detector outperforms ISS in terms of relative repeatability. Hence, it can be claimed from Figs. 3(a), (b) and 4 that HoNO keypoints are more meaningful and offer better repeatability than ISS keypoints.

Even the KPQ keypoint detector has the same problem of detecting keypoints on planar regions as it employs similar pruning method [21] as in ISS and because of the parameters used in evaluation. One can see that KPQ keypoints also lie on planar regions as shown in Fig. 8 of [21].

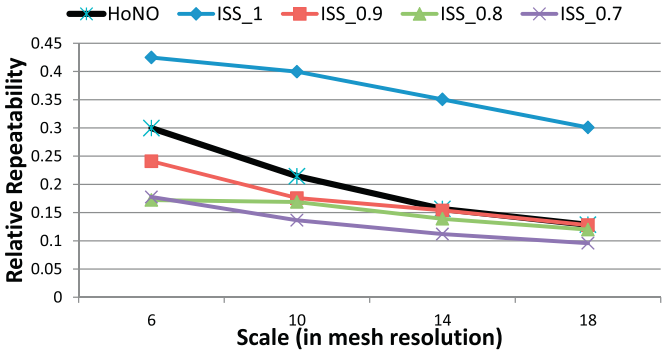


Fig. 4. Performance of ISS keypoint detector with varying Th_{32} and Th_{21} in comparison with HoNO. ISS_0.9 represents that $Th_{32} = Th_{21} = 0.9$.

In short, ISS and KPQ keypoint detectors with the employed parameters in [21], whose repeatability performance is shown in Fig. 2, do not detect meaningful and useful keypoints as most of them lie on planar regions. Moreover, by changing the parameters to detect informative keypoints on curved regions, there is significant drop in their performance and offer lower repeatability than the proposed HoNO keypoints.

3.4.2. Other keypoint detectors

The HKS keypoint detector offered good relative repeatability on StandfordViews and Retrieval datasets but its absolute repeatability highlights that it detects less number of keypoints and do not perform well on Kinect and SpaceTime datasets. As mentioned in [21], on a computer with 6 GB RAM, HKS can only run on meshes with a maximum of 30,000 vertices, which highlights its need for high computational resources.

SURE, NARF and SDTP keypoint detectors detect much lesser number of keypoints and their repeatability on these datasets is low. We tested with various parameters and it turned out that SURE keypoint detector detects ~100 keypoints on scenes from Kinect dataset and has relative repeatability less than 0.1. Wang et al. [24] have shown that SDTP outperforms NARF in relative repeatability on Kinect and SpaceTime datasets. However, SDTP's relative repeatability is still lower than ours, which can be seen

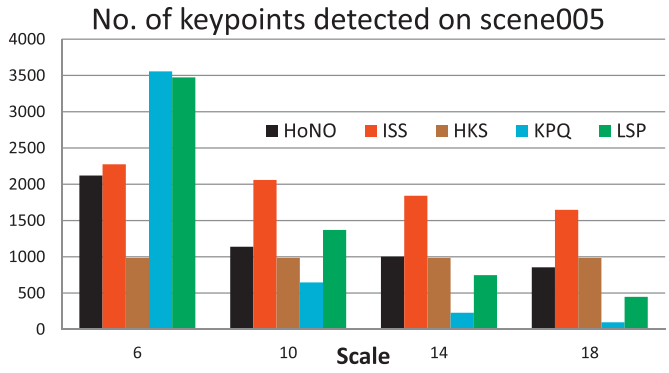


Fig. 5. Number of keypoints detected on `scene005` (shown in Fig. 1) by different keypoint detectors and their variation with scale.

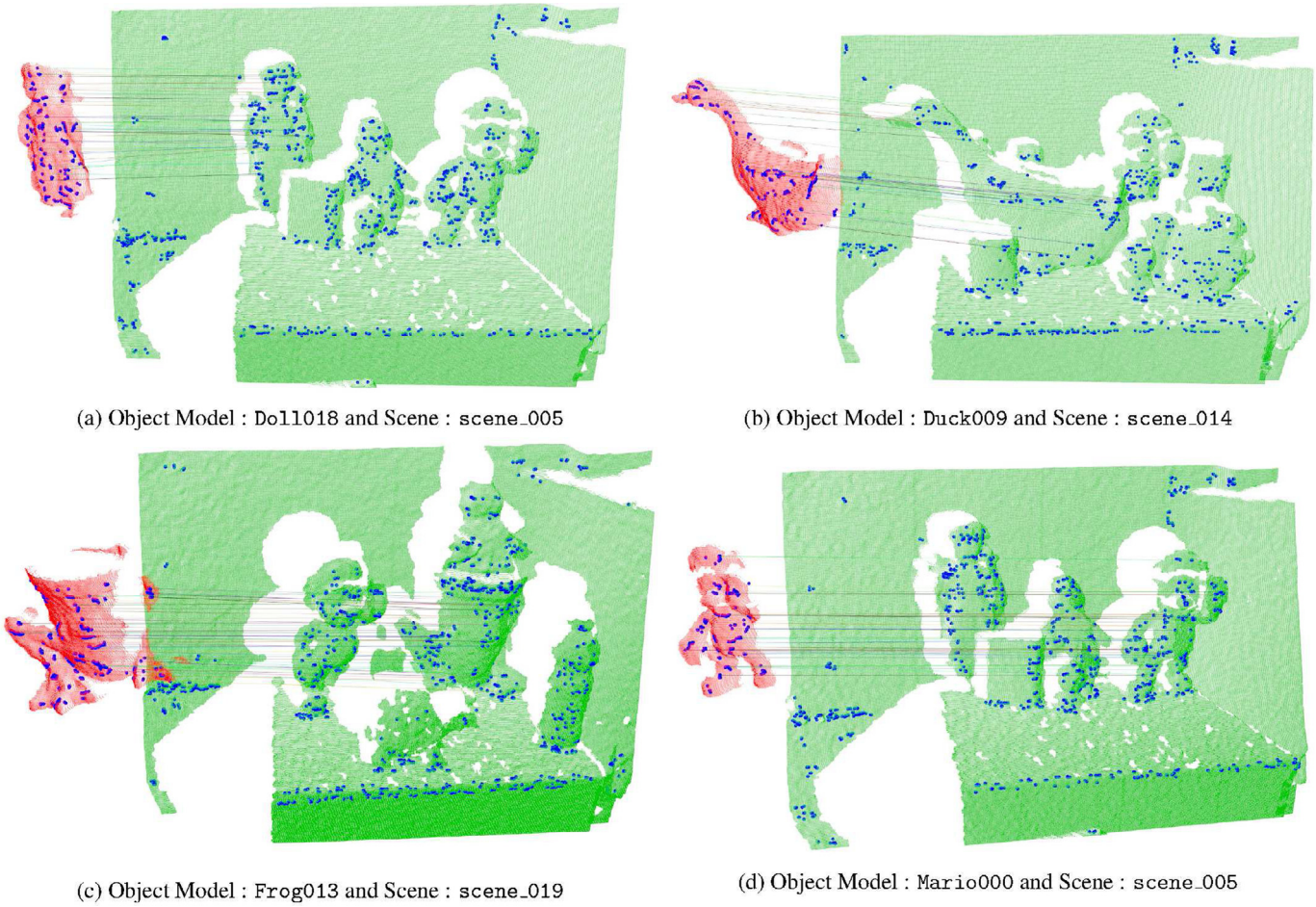
by comparing their published results on Kinect and SpaceTime datasets with ours.

One question may arise in the reader's mind that the proposed approach may detect more number of keypoints when compared to others and that might be the reason for high performance in repeatability tests. We address this question by quantitatively showing in Fig. 5 that our method detects almost the same number or sometimes lesser number of keypoints than other keypoint detectors. Specifically, in Fig. 5, we show the number of keypoints detected on a `scene005` from Kinect dataset and their variation with the scale.

In any real world application, it is highly possible to have scenes with some kind of planar background. Keypoint detectors should avoid detecting keypoints on the planar surfaces as feature descriptor based keypoint matching would become highly ambiguous. From the datasets provided by Tombari et al. [21], Kinect dataset is closer to real world applications as it has noise that arises directly from Kinect sensor and planar backgrounds that are quite common. HoNO outperforms other keypoint detectors on this dataset. Moreover it can be claimed from Figs. 2–4 of this letter and Fig. 8 of [21] that HoNO offers more reliable keypoints than other existing keypoint detectors.

3.5. HoNO keypoint sets + SHOT feature descriptor

In this experiment, we combine HoNO keypoint sets with the SHOT feature descriptor and illustrate few results on the Kinect dataset in Fig. 6. We consider various object models from the Kinect dataset such as `Doll, Duck, Frog and Mario` and scenes that contain them. HoNO keypoints are detected with a scale of 10 $\times$ mesh resolution $\approx$ 0.02m, which are then matched with SHOT feature descriptors constructed with a support size of 0.08m and RANSAC was used to remove false correspondences. It can be seen from Fig. 6 that the keypoints from the model are matched with the relevant keypoints in the scene successfully. These informative HoNO keypoints can be used with binary 3D



(a) Object Model : `Doll018` and Scene : `scene_005`

(b) Object Model : `Duck009` and Scene : `scene_014`

(c) Object Model : `Frog013` and Scene : `scene_019`

(d) Object Model : `Mario000` and Scene : `scene_005`

Fig. 6. HoNO keypoints are detected on `Doll, Duck, Frog and Mario` object models from the Kinect dataset and they are matched with SHOT feature descriptor to the keypoints detected in various scenes. In the figure, object models are shown in red, scenes are shown in green while the HoNO keypoints are shown in blue. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

feature descriptor, B-SHOT [13], as well, to achieve even faster keypoint matching.

4. Conclusion and future work

This letter introduced a new method to detect 3D keypoints on point clouds by leveraging on histogram of normal orientations that is calculated at every point. Compared to existing 3D keypoint detectors, the detected keypoints are highly repeatable, avoid planar regions effectively and lie on informative regions of 3D point clouds. The detected keypoints lie in groups and these groups are distinctive. The detection of these repeatable keypoint sets opens up new research directions such as developing a feature descriptor for these keypoint sets and to match those keypoint sets by incorporating additional information from neighboring keypoint sets.

References

[1] A. Albarelli, E. Rodola, A. Torsello, A game-theoretic approach to fine surface registration without initial motion estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2010, pp. 430–437.
[2] A. Albarelli, E. Rodola, A. Torsello, Loosely distinctive features for robust surface alignment, in: Proceedings of the IEEE Conference on Computer Vision–ECCV, Springer, 2010, pp. 519–532.
[3] H. Chen, B. Bhanu, 3D free-form object recognition in range images using local surface patches, Pattern Recognit. Lett. 28 (10) (2007) 1252–1262.
[4] C. Dorai, A.K. Jain, COSMOS-A representation scheme for 3D free-form objects, Pattern Anal. Mach. Intell. IEEE Trans. 19 (10) (1997) 1115–1130.
[5] F. Endres, J. Hess, J. Sturm, D. Cremers, W. Burgard, 3-d mapping with an rgb-d camera, Robot. IEEE Trans. 30 (1) (2014) 177–187, doi:10.1109/TRO.2013.2279412.
[6] T. Fiolka, J. Stückler, D.A. Klein, D. Schulz, S. Behnke, SURE: surface entropy for distinctive 3D features, in: Spatial Cognition VIII, Springer, 2012, pp. 74–93.
[7] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, 3D object recognition in cluttered scenes with local surface features: a survey, Pattern Anal. Mach. Intell. IEEE Trans. 36 (11) (2014) 2270–2287, doi:10.1109/TPAMI.2014.2316828.
[8] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2) (2004) 91–110.
[9] A. Mian, M. Bennamoun, R. Owens, On the repeatability and quality of keypoints for local feature-based 3D object retrieval from cluttered scenes, Int. J. Comput. Vis. 89 (2–3) (2010) 348–361, doi:10.1007/s11263-009-0296-z.
[10] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, L. Gool, A comparison of affine region detectors, Int. J. Comput. Vis. 65 (1–2) (2005) 43–72, doi:10.1007/s11263-005-3848-x.
[11] S. Prakhya, L. Bingbing, Y. Rui, W. Lin, A closed-form estimate of 3D ICP covariance, in: Proceedings of the 14th IAPR International Conference on Machine Vision Applications (MVA), 2015, pp. 526–529, doi:10.1109/MVA.2015.7153246.
[12] S. Prakhya, L. Bingbing, L. Weisi, U. Qayyum, Sparse depth odometry: 3D keypoint based pose estimation from dense depth data, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2015, pp. 4216–4223, doi:10.1109/ICRA.2015.7139780.
[13] S. Prakhya, B. Liu, W. Lin, B-SHOT: a binary feature descriptor for fast and efficient keypoint matching on 3D point clouds, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015c, pp. 1929–1934, doi:10.1109/IROS.2015.7353630.
[14] E. Rodolà, A. Albarelli, F. Bergamasco, A. Torsello, A scale independent selection process for 3d object recognition in cluttered scenes, Int. J. Comput. Vis. 102 (1) (2012) 129–145.
[15] S. Rusinkiewicz, M. Levoy, Efficient variants of the ICP algorithm, in: Proceedings of the Third International Conference on 3-D Digital Imaging and Modeling, IEEE, 2001, pp. 145–152.
[16] R. Rusu, S. Cousins, 3D is here: Point Cloud Library (PCL), in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2011, doi:10.1109/ICRA.2011.5980567.
[17] S. Salti, F. Tombari, L.D. Stefano, Shot: unique signatures of histograms for surface and texture description, Comput. Vis. Image Underst. 125 (2014) 251–264. URL: http://www.sciencedirect.com/science/article/pii/S1077314214000988
[18] C. Schmid, R. Mohr, C. Bauckhage, Evaluation of interest point detectors, Int. J. Comput. Vis. 37 (2) (2000) 151–172, doi:10.1023/A:1008199403446.
[19] B. Steder, R. Rusu, K. Konolige, W. Burgard, Point feature extraction on 3D range scans taking into account object boundaries, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2011, pp. 2601–2608, doi:10.1109/ICRA.2011.5980187.
[20] F. Tombari, S. Salti, L. Di Stefano, unique signatures of histograms for local surface description, in: Proceedings of the Computer Vision–ECCV, Springer, 2010, pp. 356–369.
[21] F. Tombari, S. Salti, L. Di Stefano, Performance evaluation of 3D keypoint detectors, Int. J. Comput. Vis. 102 (1–3) (2013) 198–220, doi:10.1007/s11263-012-0545-4.
[22] F. Tombari, S. Salti, L.D. Stefano, A combined texture-shape descriptor for enhanced 3d feature matching, in: Proceedings of the 18th IEEE International Conference on Image Processing (ICIP), IEEE, 2011, pp. 809–812.
[23] A. Torsello, E. Rodola, A. Albarelli, Sampling relevant points for surface registration, in: Proceedings of the International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), IEEE, 2011, pp. 290–295.
[24] S. Wang, L. Gong, H. Zhang, Y. Zhang, H. Ren, S.-M. Rhee, H.-E. Lee, SDTP: a robust method for interest point detection on 3d range images, 9025 (2014) 90250O-90250O-9. doi:10.1117/12.2037133.
[25] Y. Zhong, Intrinsic shape signatures: a shape descriptor for 3D object recognition, in: Proceedings of the IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops), 2009, pp. 689–696, doi:10.1109/ICCVW.2009.5457637.