






Performance Evaluation of Selected 3D Keypoint Detector–Descriptor Combinations

Paula Stancelova¹(✉) , Elena Sikudova² , and Zuzana Cernekova¹ 

¹ Faculty of Mathematics, Physics and Informatics,
Comenius University in Bratislava, Bratislava, Slovakia
paula.budzakova@fmph.uniba.sk, zuzana.cernekova@uniba.sk

² Faculty of Mathematics and Physics, Charles University,
Prague, Czech Republic
sikudova@cgg.mff.cuni.cz

Abstract. Nowadays, with easily accessible 3D point cloud acquisition tools, the field of point cloud processing gained a lot of attention. Extracting features from 3D data became main computer vision task. In this paper, we reviewed methods of extracting local features from objects represented by point clouds. The goal of the work was to make theoretical overview and evaluation of selected point cloud detectors and descriptors. We performed an experimental assessment of the repeatability and computational efficiency of individual methods using the well known Stanford 3D Scanning Repository database with the aim of identifying a method which is computationally-efficient in finding good corresponding points between two point clouds. We combine the detectors with several feature descriptors and show which combination of detector and descriptor is suitable for object recognition task in cluttered scenes. Our tests show that choosing the right detector impacts the descriptor's performance in the recognition process. The repeatability tests of the detectors show that the data which contained occlusions have a high impact on their performance. We summarized the results into graphs and described them with respect to the individual tested properties of the methods.

Keywords: 3D detector · 3D descriptor · Point cloud · Feature extraction

1 Introduction

With easily accessible 3D point cloud acquisition tools it is natural that the field of point cloud processing gained a lot of attention in the past decade. Many computer vision tasks working with 3D point clouds benefit from using 3D local features. Among possible applications of 3D local features are point cloud registration (pose estimation) [6], automatic object localization and recognition in 3D point cloud scenes [26], model retrieval [3] or face recognition [15].

The pipeline of these applications usually starts with detecting keypoints using detectors designed by studying the geometric properties of local structures [6]. The next step is the description of the keypoints and finally the classification or matching of the keypoints. The latest deep learning approach combines all three steps into one. The features and their descriptions are learned during the training of the classifier [13, 17].

When using the hand crafted features we need to ensure that the detected keypoints are representative and the descriptors are robust to clutter and occlusion in order to get reliable results (classification, localization,...).

The goal of the paper is to provide the performance evaluation of hand crafted feature detectors for point clouds. First we address the ability to detect repeatable points. Second, since the efficiency of the descriptors can influence the results, we combine the detectors with several feature descriptors and show which combination of detector and descriptor is suitable for object recognition task in cluttered scenes.

2 Related Work

Many existing works test and compare methods to extract 3D local features. Filipe and Alexandre [7] work with a real dataset of different RGB-D objects. They test 3D detectors in the Point Cloud Library (PCL) [20] that work directly on RGB-D data. They only evaluate the performance of the detectors in terms of various transformations. Tombari, Salti and di Stefano [25] perform in-depth testing of state-of-the-art detectors, which they divide into two new categories (fixed-scale, adaptive-scale). Testing is performed on 3D scene built by randomly rotating and translating the selected 3D models of synthetic data. A quantitative comparison of the performance of the three most general detectors and two descriptors for monitoring plant health and growth is given in [1]. The choice of test methods was based on their most common use in various applications. Mian, Bennamoun and Owens [16] propose a multi-scale keypoint detection algorithm for extracting scale invariant local features. They also propose a technique for automatically selecting the appropriate scale at each keypoint and extract multi-scale and scale invariant features. They test the methods for 3D object retrieval from complex scenes containing clutter and occlusion. Hänsch, Weberand and Hellwich [10] explore the benefits of 3D detectors and descriptors in the specific context of point cloud fusion that can be used, for example, to reconstruct a surface.

Recent trends in data driven approaches encouraged to employ deep learning. Representative works include PPFNet [5], PCPNet [9], 3D-FeatNet [27] and PPF-FoldNet [4]. All the mentioned works surpassed the handcrafted alternatives with a big margin. 2D features are usually supplemented by useful information about the local orientation, derived from the local visual part of the image. For 3D features, it is very difficult to find a unique and consistent local coordinate frame. None of the aforementioned works were able to attach the local orientation information to 3D patches.

Inspired by [25], we propose a comparison of 3D detectors that work primarily on the point clouds without the use of the color information. Like [25], we focus on the object recognition scenario, which is characterized mainly by a large occlusion. Subsequently, we assess the performance of the selected 3D descriptors in combination with the tested detectors because, it is not clear how different keypoints detectors and descriptors work together. Our purpose is to show the performance of selected combinations in recognizing objects in a scene, as we have not found these combinations in the existing papers.

3 3D Keypoint Detectors, Descriptors and Matching

In this section we will describe detectors and descriptors we have used in our tests together with the matching method we have used.

3.1 3D Keypoint Detectors

Harris3D. The Harris3D detector [22] is the extension of the 2D corner detection method of Harris and Stephens [11]. Harris3D is using first order derivatives along two orthogonal directions on the surface. The derivatives are obtained so that the neighboring area is approximated by quadratic surface. Afterwards, similarly to 2D detector, the covariance matrix is calculated, but the image gradients are replaced by the surface normals. For final detection of keypoints, the Hessian matrix of intensity is used for every point, smoothed by an isotropic Gaussian filter.

Intrinsic Shape Signatures 3D (ISS3D). The ISS3D keypoint detector [28] relies on measuring the local distances or the points density. It uses the eigenvalues and eigenvectors of the scatter matrix, created from supporting region points. The eigenvector corresponding to the smallest eigenvalue is used as a surface normal and the ratio between two successive eigenvalues is used to reject similar points.

Normal Aligned Radial Feature (NARF). Detector NARF [23] is designed to choose the keypoints, so that they are able to repair the information about the object borders and its surface to obtained highest robustness of the method. The selected points are supposed to be in positions where the surface is stable and where there are sufficient changes in the local neighborhood to be robustly detected in the same place even if observed from different perspectives. At first, the detector determines the object border, for which the change of distances between two adjacent points is used. To achieve the keypoint stability, for every point and its local neighborhood, the amount and dominant direction of the surface changes at this position are determined. The interest value is calculated as the difference between orientations in area and changes of the surface. Afterwards, the smoothing of the interest values and non-maximum suppression is performed to determine the final keypoints.

SIFT3D. SIFT 3D is an extension of the original 2D SIFT detector, where for finding the keypoints the scale pyramid of difference of Gaussians (DoG) is used [14]. SIFT 3D is using 3D version of Hessian matrix, where the density function is approximated by sampling the data regularly in space. Over the density function a scale space is built, where search for local maxima of the Hessian determinant is performed. The input cloud is convolved with a number of Gaussian filters whose standard deviations differ by a fixed scale factor. Using the convolution the point clouds are smoothed and the difference is calculated to obtain three to four DoG point clouds. These two steps are repeated to get higher number of DoG in the scale-space. The 3D SIFT keypoints are identified as the scale-space extrema of the DoG function similarly to 2D SIFT.

3.2 3D Local Features Descriptors

Many existing 3D local features descriptors use histograms to represent different characteristics of the local surface. Principally, they describe the local surface by combining geometric or topological measurements into histograms according to point coordinates or geometric attributes. We classify these descriptors into two groups: *geometric attribute histogram* and *spatial distribution histogram* based descriptors.

Geometric Attribute Histogram Based Descriptors. These descriptors represent local surface by generating histograms based on geometric attributes, as normals or curvatures, of the points on the surface. These descriptors also mostly construct the Local Reference Frame (LRF) and/or Local Reference Axis (LRA).

Point Feature Histogram (PFH): The PFH descriptor [19] works with point pairs in the support region. First, the Darboux frame is defined by using the surface normals and point positions, for each pair of the points in the support region. Then, four features for each point pair using the Darboux frame, the surface normals, and their point positions are calculated. The PFH descriptor is generated by accumulating points in particular bins along the four dimensions. The length of the final PFH is based on the number of histogram bins along each dimension.

Fast Point Feature Histogram (FPFH): The FPFH descriptor [18] consists of two steps. In the first step, a Simplified Point Feature Histogram (SPFH) is generated for each point by calculating the relationships between the point and its neighbors. In SPFH, the descriptor is generated by chaining three separate histograms along each dimension. In the second step, FPFH descriptor is constructed as the weighted sum of the SPFH of the keypoint and the SPFHs of the points in the support region. The length of the FPFH descriptor depends on the number of histogram bins along each dimension.

Signature of Histograms of Orientations (SHOT): The SHOT descriptor [21] originated as an inspiration by SIFT descriptor. The descriptor encodes the

histograms of the surface normals in different spatial locations. First, the LRF is constructed for each keypoint and its neighboring points in the support region are aligned with the LRF. The support region is divided into several units along the radial, azimuth and elevation axes. Local histogram for each unit is generated by accumulating point counts into bins according to the angles between the normals at the neighboring points within the volume and the normal at the keypoint. The final descriptor is chaining all the local histograms.

Spatial Distribution Histogram Based Descriptors. These descriptors represent local surface by generating histograms according to spatial coordinates. They mostly start with the construction of a LRF or LRA for keypoint, and divide 3D support region to the LRF/LRA. Afterwards, they generate a histogram for local surface by accumulating the spatial distribution measurements for each spatial bin.

3D Shape Context (3DSC): 3DSC descriptor [8] is extension of the 2D Shape Contexts descriptors [2]. The algorithm use the surface normal at the keypoint as its LRA. First, a spherical grid is placed at the keypoint, which is the center of the grid, with the north pole of the grid being aligned with the surface normal. The support region is divided into several bins, logarithmic along the radial dimension and linear along the azimuth and the elevator dimensions of the spherical grid. The 3DSC descriptor is generated by counting the weighted number of points falling into each bin of the 3D grid. The final descriptor is chaining the three partial descriptors, that represent the numbers of bins along the radial, azimuth and elevation axes.

Unique Shape Context (USC): USC descriptor [24] is extension of 3DSC, which bypasses computing multiple descriptors at a given keypoint. LRF is constructed for each keypoint and aligned with the local surface in order to provide invariance to rotations and translations. The support region is divided into several bins. Final USC descriptor is generated analogous to the approach used in 3DSC.

3.3 Descriptors Matching

In our experiments the objects in the scenes are recognized using the nearest neighbor distance ratio matching method (NNDR). In the nearest neighbor matching method, two descriptors match, if they are the closest and their distance is below a threshold. With this strategy, there is at most one match but it might not be correct, especially when there are few keypoints with similar geometric properties in the point cloud. In this case, the distances from the nearest and the second nearest descriptors are similar. To overcome this property we use the NNDR matching, where the descriptors D_A and D_B are matched if

$$\|D_A - D_B\|/\|D_A - D_C\| < t, \quad (1)$$

where D_B is the first and D_C is the second nearest neighbor to D_A and t is a specified threshold.

4 Testing

We tested the performance of detectors and descriptors separately in combination with the tested detectors. In this section we provide the evaluation criteria we have used for the tests and the results we have obtained.

4.1 Dataset

For performing our experiments, we selected four models well known in community of 3D point clouds researchers (the Stanford Bunny, the Happy Budha, the Dragon, the Armadillo) from the publicly available 3D model database the Stanford 3D Scanning Repository [12]. All the selected models were scanned by a Cyber-ware 3030 MS scanner. For the testing we used the models individually and scaled them using the scale of 6, 9, 12, 15 times the cloud resolution. We also combined several transformed models to create 15 testing scenes having 265–290 thousands of points. The scenes were created so that different parts of individual models were occluded. The occlusion was from 30% to 60%.

4.2 Evaluation Criteria

A keypoint can be useful for identifying an object, when it is detectable under changing conditions including viewpoint change, noise, partial neighborhood occlusion, etc. This property is called repeatability.

A keypoint is repeatable, if it is found in a small neighborhood of the expected location in the scene. Let k_M be a keypoint found on the original model. The expected location in the scene is then $T(k_M)$, where $T()$ is the transformation of the original model (rotation and translation) when inserted into the scene. Let k_S be the keypoint in the scene, which is the closest to $T(k_M)$. The keypoint k_M is repeatable, when

$$\|T(k_M) - k_S\| < \varepsilon. \quad (2)$$

As in [25] we evaluate the *absolute* and *relative repeatability* of the keypoints. When R is the set of repeatable keypoints found on a given model in a given scene, the absolute repeatability is the number of the repeatable keypoints

$$r_a = |R| \quad (3)$$

and the relative repeatability is measured relatively to the number of keypoints found on the model in the scene having a corresponding keypoint on the original model (K_S)

$$r_r = \frac{|R|}{|K_S|}. \quad (4)$$

While testing the detectors, the scale was set to 6, 9, 12, 15 times the cloud resolution. The results are averaged over the scales and the scenes.

The *descriptiveness* of selected descriptors is evaluated by combining them with the tested detectors. The purpose is to demonstrate the effect of different detectors on the performance of the feature descriptors. The performance of each

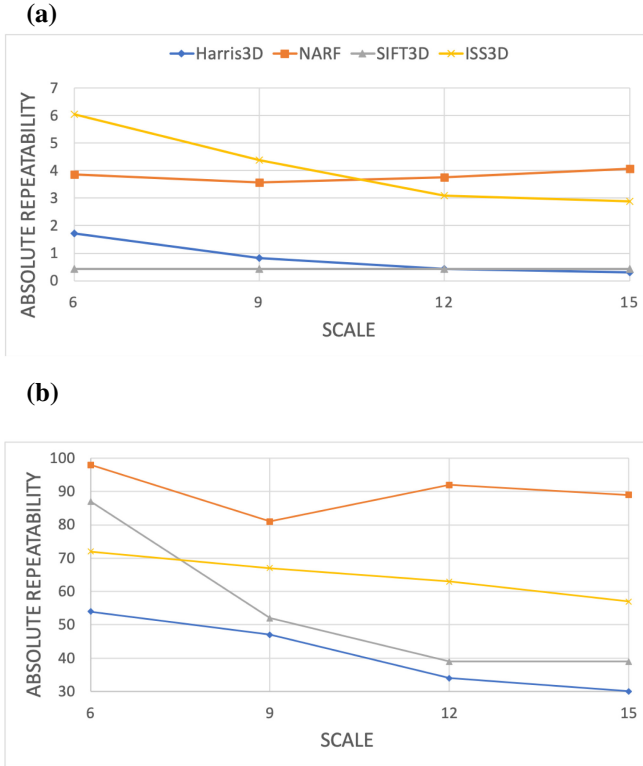


Fig. 1. (a) The average absolute repeatability of tested detectors with 60% occlusion in the scene (b) The average absolute repeatability of tested detectors with 30% occlusion in the scene

detector-descriptor combination is also measured by the Precision and Recall generated as follows. First, the keypoints are detected from all the scenes and all the models. A feature descriptor is then computed for each keypoint (not just repeatable points) using the selected descriptors. Next, we use NNDR to perform feature matching of the models with the scenes. If the distance between the pair of keypoint descriptors is less than half of the support radius, which is the threshold, the match is assumed correct. Otherwise, it is a false match. The *precision* is calculated as

$$\text{precision} = \frac{\text{Nr. of correct matches}}{\text{Nr. of identified matches}}. \quad (5)$$

The *recall* is calculated as

$$\text{recall} = \frac{\text{Nr. of correct matches}}{\text{Nr. of corresponding features}}. \quad (6)$$

We also evaluate the *computational effectiveness* of the detectors and descriptors. We calculate the time needed for detecting the keypoints on several objects

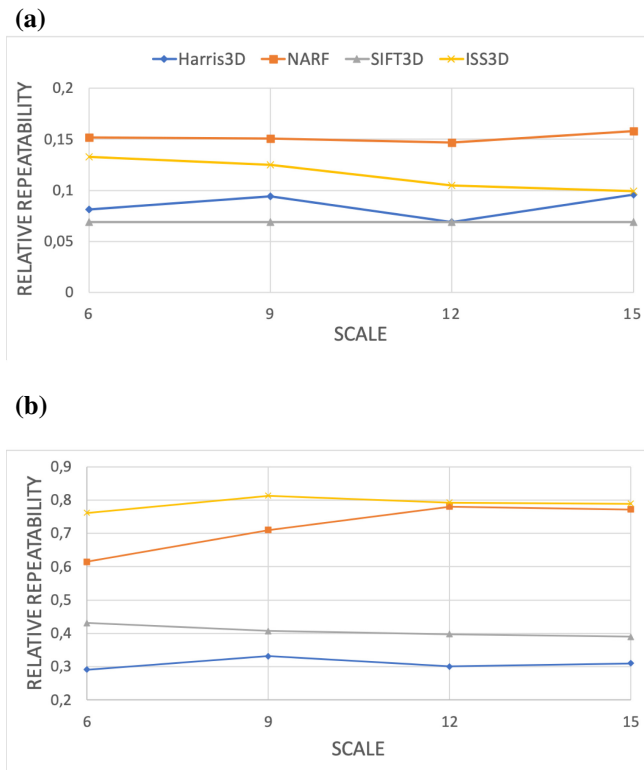


Fig. 2. (a) The average relative repeatability of tested detectors with 60% occlusion in the scene (b) The average relative repeatability of tested detectors with 30% occlusion in the scene

or scenes with various point density to determine the effectiveness of detector. Also, we calculate the time needed to describe the extracted keypoints.

All methods are used as implemented in the Point Cloud Library (PCL), which is an open-source library of algorithms for 2D/3D image and point cloud processing. For all the selected detectors, we use the default parameters proposed in the original publications.

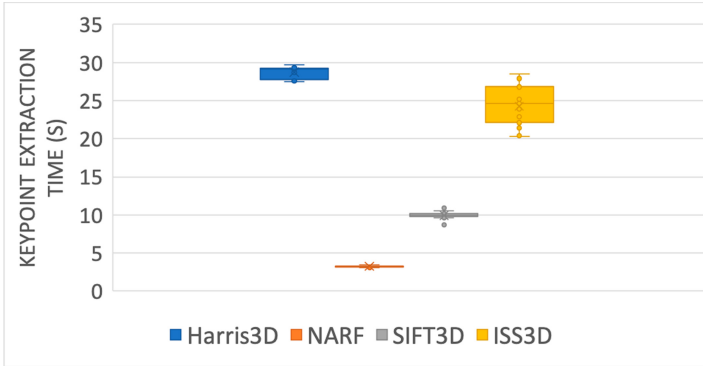


Fig. 3. Time performance of the detectors. Scale is kept fixed to 6. The scenes have 265–290 thousands of points

4.3 Results

Detectors. Based on the proposed experiments and their results we can draw interesting conclusions. In Figs. 1 and 2, we present the results of absolute and relative repeatability of HARRIS3D, NARF, SIFT3D and ISS3D detectors. To test the effect of occlusion in scenes on the detector performance, we investigate scenes that contain highly and low occluded test models. For scenes with high occlusion, up to 60% of each object is occluded, whereas the low occlusion scenes contain objects with 30% occlusion. The repeatability rates in the scenes with high occlusion are very low. Nevertheless, we can see that NARF and ISS3D detectors achieve higher rates than HARRIS3D and SIFT3D in both scenarios. The only exception is the absolute repeatability rate of SIFT3D for scale 6 in low occlusion scenes.

In the highly occluded scenes is the relative repeatability of all detectors very low. An the average number of detected keypoints does not exceed 6.

In the scenes with lower occlusion the relative repeatability reaches 80% for NARF and ISS3D detectors, which means that the points are highly repeatable, while the absolute number of detected points is also high. For HARRIS3D and SIFT3D even though the absolute numbers of detected keypoints is high for scales 6 and 9, the relative repeatability rate reaches only 40%.

The error-bar chart in Fig. 3 reports the measured detection time (in seconds) of methods computed on each model. Because all methods are implemented in the same framework, their performance can be compared by taking the average detection time on the same set of objects. The detection time is almost constant over scenes with varying number of points. The only exception is the ISS3D, where the trend is constant with higher variance.

Descriptors. In Fig. 4 we present the precision and recall measures of the descriptor matching task on scale 6 objects. All detector–descriptor combinations achieve more than 65% recall, and 50% precision (except SIFT3D–USC with 38%). We can see that the USC descriptor causes the worst performance regardless of the detector and the SIFT3D detector is the weakest from all when combined with any descriptor. The best performance was achieved by the FPFH descriptor for all the detectors. The performance of PFH, 3DSC and SHOT descriptors varies with the used detectors. The order of decreasing performance is PFH, SHOT and 3DSC descriptors for HARRIS3D, NARF and ISS3D detectors and the other way round for the SIFT3D detector. The best performance overall is achieved by the pair ISS3D–FPFH.

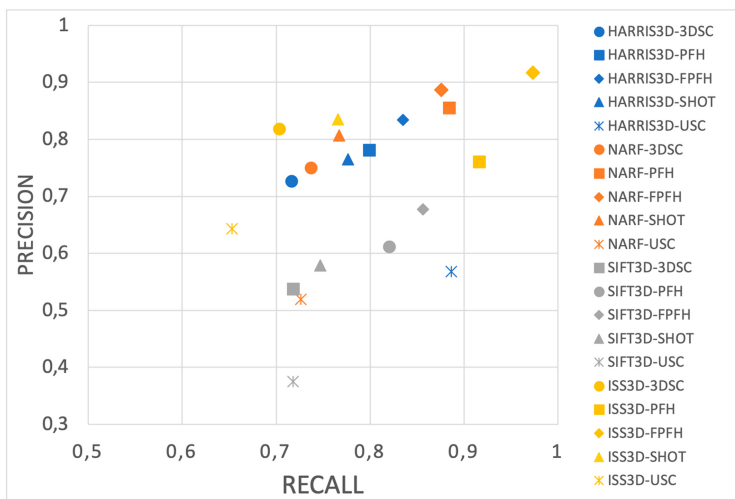


Fig. 4. Performance of the selected descriptors combined with different 3D keypoint detectors

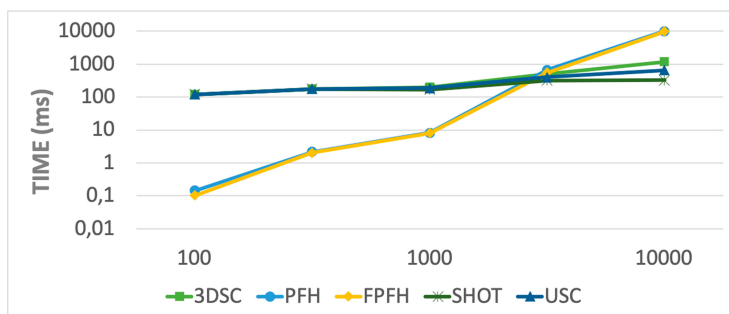


Fig. 5. Average time to generate a feature descriptor for a local surface with different number of points in the support region. The scale of both axes is logarithmic

The graph in Fig. 5 reports the measured average time (in milliseconds) of keypoint description according to the number of points in its local region. As with the detectors, the descriptors are implemented in the same framework, so their performance can be compared by taking the average time needed to generate the descriptor of the same model represented in different scales.

We can see that for small-sized support regions (100–1000 points) the PFH and FPFH descriptors are orders of magnitude faster than the other descriptors, but for 10000, they are 10 times slower. All descriptors are comparably fast for neighborhoods with around 5000 points.

5 Conclusions

This paper presents the performance evaluation of hand crafted feature detectors for point clouds. The repeatability tests of the detectors show that the data which contained occlusions have a high impact on their performance. The best repeatable detector of the set of test detectors in scale 6 is ISS3D. For the scenes with lower occlusion are the best repeatable NARF and ISS3D detectors. The best detector–descriptor pairs performance for object recognition scenario is achieved by ISS3D–FPFH combination. Our tests also show that choosing the right detector impacts the descriptor’s performance in the recognition process. The FPFH and PFH descriptors are the fastest in time measurements for small number of local region’s point. At higher points, the descriptors measurements are comparable.

Acknowledgment. This work has been funded by Slovak Ministry of Education under contract VEGA 1/0796/00 and by the Charles University grant SVV-260588.

References

1. Azimi, S., Lall, B., Gandhi, T.K.: Performance evaluation of 3D keypoint detectors and descriptors for plants health classification. In: 2019 16th International Conference on Machine Vision Applications (MVA), pp. 1–6. IEEE (2019)
2. Belongie, S., Mori, G., Malik, J.: Matching with shape contexts. In: Krim, H., Yezzi, A. (eds.) *Statistics and Analysis of Shapes*, pp. 81–105. Springer, Heidelberg (2006). https://doi.org/10.1007/0-8176-4481-4_4
3. Bold, N., Zhang, C., Akashi, T.: 3D point cloud retrieval with bidirectional feature match. *IEEE Access* **7**, 164194–164202 (2019)
4. Deng, H., Birdal, T., Ilic, S.: PPF-FoldNet: unsupervised learning of rotation invariant 3D local descriptors. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) *ECCV 2018*. LNCS, vol. 11209, pp. 620–638. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01228-1_37
5. Deng, H., Birdal, T., Ilic, S.: PPFNet: global context aware local features for robust 3D point matching. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 195–205 (2018)
6. Deng, H., Birdal, T., Ilic, S.: 3D local features for direct pairwise registration. In: *Computer Vision and Pattern Recognition (CVPR)*. IEEE (2019)

7. Filipe, S., Alexandre, L.A.: A comparative evaluation of 3D keypoint detectors in a RGB-D object dataset. In: 2014 International Conference on Computer Vision Theory and Applications (VISAPP), vol. 1, pp. 476–483. IEEE (2014)
8. Frome, A., Huber, D., Kolluri, R., Bülow, T., Malik, J.: Recognizing objects in range data using regional point descriptors. In: Pajdla, T., Matas, J. (eds.) ECCV 2004. LNCS, vol. 3023, pp. 224–237. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24672-5_18
9. Guerrero, P., Kleiman, Y., Ovsjanikov, M., Mitra, N.J.: PCPNet learning local shape properties from raw point clouds. In: Computer Graphics Forum, vol. 37, pp. 75–85. Wiley Online Library (2018)
10. Hänsch, R., Weber, T., Hellwich, O.: Comparison of 3D interest point detectors and descriptors for point cloud fusion. ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. **2**(3), 57 (2014)
11. Harris, C.G., Stephens, M., et al.: A combined corner and edge detector. In: Proceedings of the 4th Alvey Vision Conference, pp. 23.1–23.6 (1988)
12. Levoy, M., Gerth, J., Curless, B., Pull, K.: The Stanford 3D scanning repository (2005). <http://www-graphics.stanford.edu/data/3Dscanrep>
13. Li, J., Chen, B.M., Lee, G.H.: SO-Net: Self-organizing network for point cloud analysis. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9397–9406, June 2018
14. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 2, pp. 1150–1157. IEEE (1999)
15. Markuš, N., Pandžić, I., Ahlberg, J.: Learning local descriptors by optimizing the keypoint-correspondence criterion: applications to face matching, learning from unlabeled videos and 3D-shape retrieval. IEEE Trans. Image Process. **28**(1), 279–290 (2019)
16. Mian, A., Bennamoun, M., Owens, R.: On the repeatability and quality of keypoints for local feature-based 3D object retrieval from cluttered scenes. Int. J. Comput. Vis. **89**(2–3), 348–361 (2010). <https://doi.org/10.1007/s11263-009-0296-z>
17. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: PointNet++: deep hierarchical feature learning on point sets in a metric space. In: Guyon, I., et al. (eds.) Advances in Neural Information Processing Systems, vol. 30, pp. 5099–5108. Curran Associates, Inc. (2017)
18. Rusu, R.B., Blodow, N., Beetz, M.: Fast point feature histograms (FPFH) for 3D registration. In: 2009 IEEE International Conference on Robotics and Automation, pp. 3212–3217. IEEE (2009)
19. Rusu, R.B., Blodow, N., Marton, Z.C., Beetz, M.: Aligning point cloud views using persistent feature histograms. In: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3384–3391. IEEE (2008)
20. Rusu, R.B., Cousins, S.: 3D is here: Point Cloud Library (PCL). In: IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, 9–13 May 2011
21. Salti, S., Tombari, F., Di Stefano, L.: SHOT: unique signatures of histograms for surface and texture description. Comput. Vis. Image Underst. **125**, 251–264 (2014)
22. Sipiran, I., Bustos, B.: Harris 3D: a robust extension of the Harris operator for interest point detection on 3D meshes. Vis. Comput. **27**(11), 963 (2011). <https://doi.org/10.1007/s00371-011-0610-y>
23. Steder, B., Rusu, R.B., Konolige, K., Burgard, W.: Point feature extraction on 3D range scans taking into account object boundaries. In: 2011 IEEE International Conference on Robotics and Automation (ICRA), pp. 2601–2608. IEEE (2011)

24. Tombari, F., Salti, S., Di Stefano, L.: Unique shape context for 3D data description. In: Proceedings of the ACM Workshop on 3D Object Retrieval, pp. 57–62 (2010)
25. Tombari, F., Salti, S., di Stefano, L.: Performance evaluation of 3D keypoint detectors. *Int. J. Comput. Vis.* **102**(1–3), 198–220 (2013). <https://doi.org/10.1007/s11263-012-0545-4>
26. Vargas, J., Garcia, A., Oprea, S., Escolano, S., Rodriguez, J.: Object recognition pipeline: Grasping in domestic environments, pp. 18–33. IGI Global (2018)
27. Yew, Z.J., Lee, G.H.: 3DFeat-Net: weakly supervised local 3D features for point cloud registration. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11219, pp. 630–646. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01267-0_37
28. Zhong, Y.: Intrinsic shape signatures: a shape descriptor for 3D object recognition. In: 2009 IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops), pp. 689–696. IEEE (2009)