Local Point Pair Feature Histogram for Accurate 3D Matching

Anders Glent Buch
anbu@mmmi.sdu.dk
Dirk Kraft
kraft@mmmi.sdu.dk

SDU Robotics
University of Southern Denmark
Odense, DK

Abstract

This paper reports the discovery of a fast, yet highly discriminative local 3D descriptor for point cloud data. Local descriptors are popular and highly effective for various 3D tasks such as registration, pose estimation and object recognition. Good solutions for these tasks critically depend on the ability to make correct associations between two or more models, or the local features on these, even under the influence of disturbances such as noise, clutter and occlusions. Our descriptor formulation is inspired by the geometric relations employed by the well-known Point Pair Feature, used originally on a global scale for classification and later on a semi-global scale for recognition. We have identified the most discriminative subset of relations for use in a local descriptor, resulting in a condensed representation of the local variation around a surface point.

We compare against seven competing mesh and point cloud descriptors on eight different matching benchmarks with a well-defined evaluation protocol. In all cases, our descriptor outperforms earlier works, providing relative gains in accuracy above 100 % for two of the four real datasets considered. Finally, we subject all descriptors to RANSAC based pose estimation and object recognition evaluation on four real datasets. In all four cases, our descriptor matches or surpasses state of the art performances.

1 Introduction

3D processing tasks such as object recognition, detection and localization often require robust methods in order to cope with real sensor data, where imperfect reconstructions are present. These considerations are relevant for *e.g.* robotic tasks, where objects need to be grasped and manipulated. This work deals with the problem of robustly describing known, rigid objects using 3D sensor data. This 3D sensor data can come from for example laser scanning, time of flight measurements, stereo cameras or RGB-D cameras.[1] In most cases, an object model is given either as CAD data or from an object scanning process. The objective is to instantiate this model into a scene view containing partial data from the object as well as a large amount of spurious data from other elements of the scene. Additionally, the object data in the scene is both inaccurate due to sensor noise and discretization and sometimes inadequate due to missing reconstructions, *i.e.* holes, occlusion.

[1] In our work we are interested in the pure geometric information, thus we would not benefit from the additional appearance information RGB-D delivers, but our descriptor can be used to describe data from all of these sensor sources. We have tested our descriptor on data from synthetic, laser scanner, RGB-D and stereo sources.

Local 3D descriptors provide a means to get get putative, point-wise correspondences between the object model and the scene data. These can be fed to a robust estimator such as RANSAC [8] to get an estimate of the object pose, thus providing an instance recognition in the scene. Other relevant applications using local 3D descriptor matches are object reconstruction, scene registration and face recognition. Clearly, the performance of all these approaches greatly depends on the number of inlier correspondences produced by the matching stage, which amounts to a high-dimensional nearest neighbor search between two sets of descriptors.

In this work, we contribute with a new local point cloud descriptor for obtaining correspondences between 3D models of objects and scenes. We use low-level relations between oriented point pairs within a local spherical support, which makes our descriptor among the fastest available. We provide experiments, following the protocol of several prior works for benchmarking local 3D descriptors, on eight different matching tasks, both from synthetic and real datasets. Finally, we include a recognition experiment on the real datasets. Our experiments indicate substantial improvements over existing descriptors for the task of local 3D descriptor matching. A C++ implementation of our descriptor is available at `gitlab.com/caro-sdu/covis`.

In the following section, we outline the most relevant and recent related work on local 3D description. In Sect. 3 we describe the method for constructing our descriptor. All experimental results are given in Sect. 4, and in Sect. 5 we conclude on our findings.

2 Related work

Perhaps the earliest work on 3D object recognition based on local 3D features is due to Stein and Medioni [22], who defined a local representation of differential surface properties. Chua and Jarvis [5] instead used point information to derive a point signature for matching range images. Johnson and Hebert introduced the well-known Spin Image descriptor [13], also for use in object recognition from range data. Some of the following works on 3D matching and recognition showed improvements over this descriptor, *e.g.* for car detection [9], object recognition [16] and face recognition [17]. In the years that followed, an array of new 3D descriptors for both mesh and point cloud data were introduced. In [1], local depth maps were used for computing scale-invariant local descriptors. Local surface patches with high variations were described by 2D histograms in [4]. In [20] a fast and relatively low-dimensional descriptor was built using three dihedral angles. Variable-sized descriptors were used in an optimization platform in [23]. In [27] mesh data were used for keypoint detection and local description, utilizing different surface characteristics. In [24] a robust local reference frame was used to improve an existing descriptor, which immediately led to the introduction of a new and further improved point cloud based descriptor better utilizing the orientation information in the reference frame [21]. Similar principles for computing reference frames were adopted to mesh data in [10]. More recently, lower-dimensional point cloud based descriptors have shown competitive performances for some datasets [3, 14].

The descriptor presented in this work is constructed using every point pair within a local support region, resulting in a 2D histogram of distance and angle relations. Other works such as [4, 10, 13] also use 2D histograms as building blocks for descriptors, but all in different ways. We utilize some of the relations initially employed in [26] for object classification in 3D data. Similar relations were used in a notable work [7] for object recognition and pose estimation in cluttered scenes, leading to several very recent extensions [2, 6, 12]. These works

used the concept of a Point Pair Feature (PPF) as a primitive but fast method for obtaining candidate poses between two models. None of these works, however, has considered using the PPF relations for local surface description as we do, since both [26], [7] and derivatives use them globally, either in a single, global descriptor, or as a low-dimensional pair feature.

3 Method

In this section we go through the steps required to obtain a description of an object or a scene model using a set of local descriptors. The first step is to prepare the model data to ensure we have a coarse and a fine point cloud at a fixed resolution (Sect. 3.1). In Sect. 3.2 we then describe the smoothing of surface normals used. The next step, the actual creation of the 2D histogram descriptor is explained in Sect. 3.3 while the last subsection (Sect. 3.4) details how to match feature descriptors.

3.1 Data preparation

All input models for the objects and the scenes are given as 3D models such as polygon meshes or point clouds. These can either be created artificially using CAD, or they can come from laser scanners or RGB-D sensors. Before using the models we preprocess them in the following three steps: (1) First we downsample the models to a fixed resolution. Depending on the fidelity of the original data, we downsample models to a point spacing between 0.5 mm and 3 mm. We term this fine model the surface model $\mathcal{M}_s$. (2) For this model we estimate the local surface normal at each point using either the incident mesh faces (when available) or by a least squares plane fit using all points in a small neighborhood [19]. (3) Next we need to compute a set of local features to describe the model. Similar to previous benchmarks on local 3D features [3, 11, 21], we further downsample the surface model to a resolution that results in approximately 1000 remaining points on the model. We denote this coarse model the feature model $\mathcal{M}_f$. This model contains the *feature points* for which we compute local descriptors, but using the underlying points in $\mathcal{M}_s$ when describing each feature point.

All downsamplings are performed uniformly to ensure that the whole surface is covered by each descriptor. Although keypoint detectors can be used for finding more distinctive regions [25], these detectors would make our evaluations less neutral, since different keypoints can boost the performance of the descriptors to varying degrees on different datasets. The relative performances between the different descriptors are, however, to a large degree preserved when using both uniform sampling and keypoint detectors. We therefore opted for uniform samples, as it is done in earlier works [3],

3.2 Reference axis estimation

Similar to most other local 3D features, our feature formulation uses the surface normals in a local neighborhood to build a descriptor. We denote the current oriented point in $\mathcal{M}_f$ to be described as (p,n), where p is the point coordinate and n is the unit normal vector. Likewise, we denote any other oriented point in the local neighborhood in $\mathcal{M}_s$ as (p',n'). For the purpose of occlusion estimation, which will be described in the following subsection, we now estimate a stable reference axis for the feature point p. This is done by computing the average orientation of all normals within a small *internal radius*. The smoothed normal can be seen as a reference axis for our descriptor. All normals n' in the neighborhood are retained,

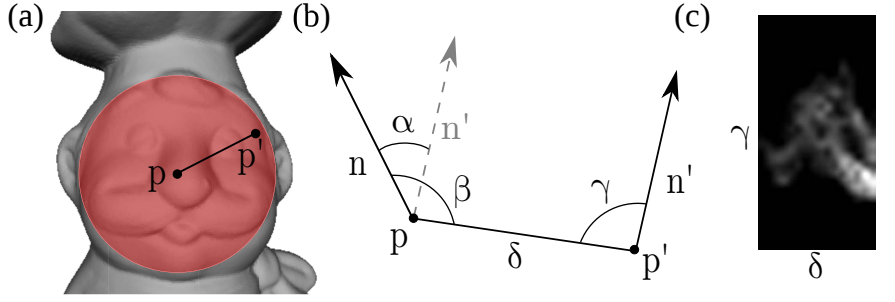


Figure 1: The PPF histogram. (a) A PPF histogram at location p describes the relation to all points p' within a radius r (overlaid red here). (b) Diagram showing the usual four PPF relational components (δ, α, β and γ) for the center point (p,n) and a point (p',n'). (c) $\delta\gamma$ PPF histogram created at the location p shown in (a).

since a smoothing of these would cause too much information loss, especially in regions with fine details. The use of a reference axis is inspired by earlier works such as [14, 21], which, however, used multiple reference axes to build a descriptor. The internal radius of our descriptor is always set to 10 % of the full radius used during histogram computation.

3.3 Histogram computation

To describe an oriented feature point (p,n), our method proceeds by finding all oriented surface points (p',n') within a spherical Euclidean neighborhood constrained by a support radius r. The search for these neighbors in $\mathcal{M}_s$ is sped up using k-d trees provided by the FLANN library [18]. The original work on PPFs [26] defined four simple geometric relations between (p,n) and (p',n') as follows (see Fig. 1 for a visualization):

$$\delta \equiv \|p'-p\| \qquad \alpha \equiv \angle(n,n') \qquad \beta \equiv \angle(n,p'-p) \qquad \gamma \equiv \angle(n',p'-p) \tag{1}$$

Depending on the fidelity of the surface model, we now have in the hundreds or thousands of $(\delta,\alpha,\beta,\gamma)$-tuples within the radius r, which will be used for describing the feature point p. To condense the information into a compact and descriptive representation, we bin the observations into histograms. The original work on PPFs [26] used a 4D histogram with $5^4 = 625$ bins for describing complete models. We tried several binning strategies, including the original 4D binning, four individual 1D binnings and various 2D binnings. Similar to other competing descriptors operating on a local scale [3, 10, 13], we have achieved the best results using 2D binnings.

To our surprise, we found that the angle γ was by far the most discriminative relation for capturing local information, while the angles α and β did not contribute to any increase in matching rate. We believe that the poor discriminative abilities of these two relations can be explained as follows. The α relation is a pure relative normal cosine relation, which encodes smoothly changing surface orientations without regard to the direction to the neighbor points, encoded in $p'-p$. On the other hand, β uses this direction information, but always with respect to the same normal vector n. Consequently, this relation neglects the orientation information provided by the neighbors in n'. The γ relation provides the best descriptive power as it integrates both types of information (the direction information neglected by α and the neighbor orientation neglected by β) in a single relation. After much initial testing, we thus ended up using a 2D histogram in the $\delta\gamma$-plane for our descriptor.

Finally, we implicitly incorporate the reference axis and the α relation for occlusion estimation. To preserve the locality of our feature, it is important to avoid neighbor points with opposing normals, since in a real scene only one side of an object is visible at a time. This is enforced by the constraint $\alpha \geq 0$. Our initial evaluations have shown this constraint to be an important component of our descriptor to achieve robustness. Indeed, without occlusion reasoning, the matching between complete object models and incomplete scene data significantly degrades. Moreover, the smoothing of the reference axis (see Sect. 3.2) used for this occlusion reasoning increases matching rates further.

The main design parameters of our descriptor are the support radius r and the number of bins along the axes of the 2D histogram. The default setting of our descriptor is to use 16 distance bins (denoted $N_\delta = 16$) and 32 bins for the angle relation (denoted $N_\gamma = 32$). With this setting, the total number of components, denoted N, in our descriptor is $N = 16 \cdot 32 = 512$. Both the feature radius and the number of histogram bins are tested in the sensitivity analysis in Sect. 4.2.

3.4 Descriptor matching

The last step in our local descriptor matching pipeline before the actual recognition stage is a matching process, where feature point correspondences between the object and scene models are obtained. A common approach is to use the L_2 metric to find the nearest Euclidean neighbors [9, 10, 13, 14, 20, 21]. The pairwise distance between a histogram descriptor on the object $H_\mathcal{O}$ and in the scene $H_\mathcal{S}$ under this metric is:

$$d_{L_2}(H_\mathcal{O}, H_\mathcal{S}) = \sqrt{\sum_{i=1}^{N} (H_{\mathcal{O},i} - H_{\mathcal{S},i})^2} \tag{2}$$

where the subscript i is used to denote the i'th component of the descriptor. We have tested this and several other distance measures, including L_1, L_∞, the Kullback-Leibler divergence, the histogram intersection kernel and the χ^2 distance (more detail on these metrics can be found in [26]). The method of choice for our PPF histograms is the χ^2 distance, defined as:

$$d_{\chi^2}(H_\mathcal{O}, H_\mathcal{S}) = \sum_{i=1}^{N} \frac{(H_{\mathcal{O},i} - H_{\mathcal{S},i})^2}{H_{\mathcal{O},i} + H_{\mathcal{S},i}} \tag{3}$$

Note that [26] achieves better results with the asymmetric version of the χ^2 distance, presumably because there is a clear notion of a reference model in their retrieval application. Contrarily, we get better matching rates with the symmetric version above, most likely because we perform a many-to-many association between local shape regions. Unlike *e.g.* the L_2 metric, the χ^2 distance explicitly accounts for the uncertainty of the individual comparisons and compared to this distance measure, χ^2 improved the matching rate of our descriptor by 5 % to 10 %, depending on the dataset. The sensitivity analysis in Sect. 4.2 includes a comparison between different metrics.

4 Results

This section presents our results. We first give, in Sect. 4.1, an extensive set of matching results for measuring the ability of our descriptor to discriminate between local structures.

Next we present a sensitivity analysis for the two main parameters of interest, the support radius and the histogram bins, as well as for the matching metric in Sect. 4.2. And finally we motivate the use of our descriptor for 3D object recognition and pose estimation on four real datasets in Sect. 4.3.

All our evaluations are performed against seven descriptors, representing both classical and new methods for local surface description. These are ECSAD [14], FPFH [20], NDHist [3], RoPS [10], SHOT [21], Spin Images (SI) [13] and USC [24].

4.1 Matching

Previous works on local feature matching have established a comprehensive set of benchmarks and a consistent evaluation protocol [3, 11, 21], which we followed in our tests.

For evaluation our contribution, we first considered the synthetic Bologna dataset of [21], which consists of a set of models and synthetic scenes, all based on models from the Stanford scanning repository[2]. There are 45 scenes containing between three and five object models in random configurations. The scenes are then corrupted by two noise levels (Bologna 1) and two decimation levels (Bologna 2), giving four variations of the whole dataset.

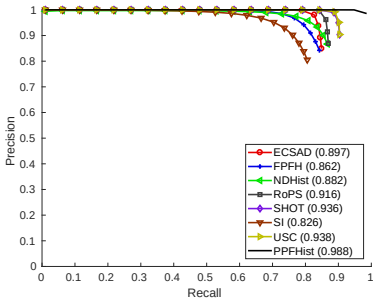
In addition to the synthetic dataset, the authors of [21] introduced two real datasets, one captured by a Kinect sensor and another generated by a spacetime stereo setup. In addition to these two datasets, we include two well-known 3D recognition datasets, UWA [16] and Queen's [23], which have also been used for previous feature evaluations, *e.g.* in [11]. For all matching experiments, we rank all feature matches using the distance ratio of the prescribed metric, which for our descriptor is the χ^2 distance. Collecting all matches in a single dataset in a sorted list allows us to compute precision-recall curves as well as maximum F_1 scores along the curves to get a single measure of accuracy. Let P and R denote precision and recall, respectively. The F_1 score is then defined as the harmonic mean of the two:

$$F_1 = 2 \cdot \frac{P \cdot R}{P + R} \tag{4}$$

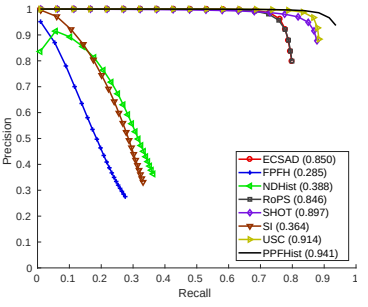
The results for all eight datasets are shown in Fig. 2 with the maximum F_1 scores along the curves shown in parentheses. From the results on the synthetic scenes in Fig. 2, our descriptors shows a high degree of robustness towards both noise and subsampling. Some descriptors suffer greatly under noise (FPFH, NDHist and SI), while subsampling also causes problems to some of the descriptors based on reference frames (SHOT and USC). The relative improvement using our descriptor over the next best descriptor ranges from 5.33 % (relative to USC at a noise level of 0.1) to 13.5 % (relative to RoPS at a resolution of 50 %).

The gap down to previous works is even higher when considering the real datasets in Fig. 2, where quite significant improvements are achieved. For the UWA, Queen's, Spacetime and Kinect the respective improvements (in terms of F_1 score) over the second best descriptor in each case are: 147 %, 203 %, 48.5 % and 33.3 %. These four datasets represent a variety of sensor characteristics going from high accuracy (UWA) to low accuracy (Kinect) and with varying levels of noise and missing data points (holes). We note that our results for the competing descriptors corroborate the most recent evaluations done in [3, 11].

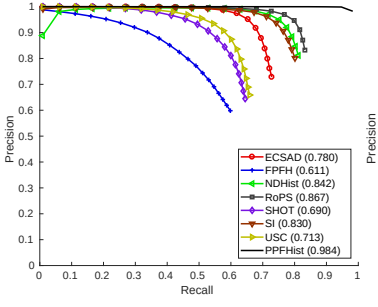
[2] http://graphics.stanford.edu/data/3Dscanrep



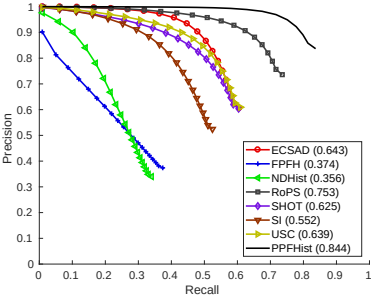
(a) Bologna 1 (noise level of 0.1).



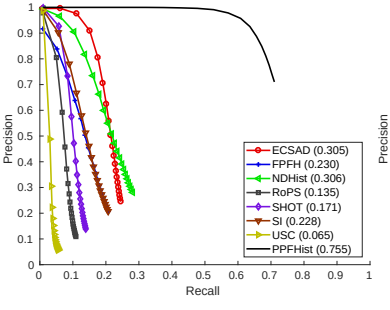
(b) Bologna 1 (noise level of 0.3).



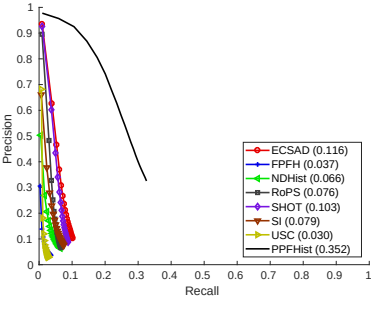
(c) Bologna 2 (resolution of 50 %).



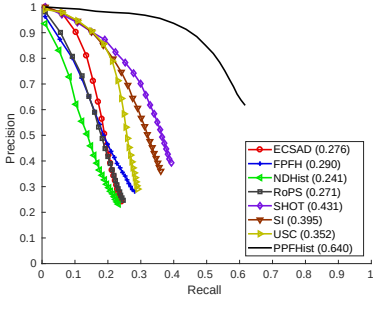
(d) Bologna 2 (resolution of 12.5 %).



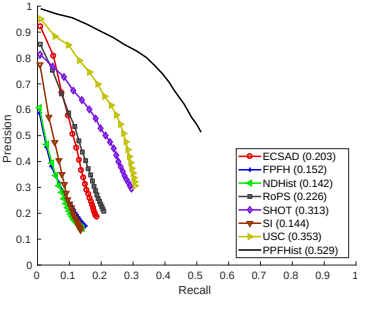
(e) UWA.



(f) Queen's.



(g) Spacetime.



(h) Kinect.

Figure 2: Matching results for the synthetic (top four) and real (bottom four) datasets. F_1 scores are shown in parentheses.

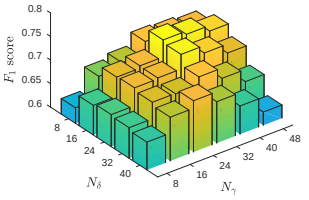


Figure 3: F_1 scores for different bin numbers.

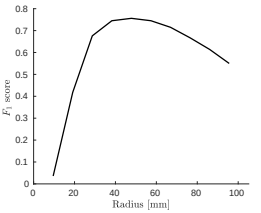


Figure 4: F_1 scores for different support radii.

Distance	F_1 score
χ^2 ratio	0.755
L_1 ratio	0.751
$\cap$	0.731
L_2 ratio	0.670
KL ratio	0.339
L_∞ ratio	0.210

Table 1: F_1 scores for different distance measures.

4.2 Sensitivity analysis

In this section, we further analyze the behavior of our descriptor under varying settings of the number of histogram bins, the local support radius and the matching metric. These tests were performed on the Bologna and UWA datasets, but similar figures are obtained when running these analyses on other datasets.

In Fig. 3 we report the influence of different binnings on the performance in terms of F_1 score on the UWA dataset. An observable peak appears for 16 distance bins and 32 angle bins. In general, more angular bins are better up to a certain point, especially if many distance bins are also used. This is likely because of instabilities in determining the correct bin for a single $\delta\gamma$ pair.

Fig. 4 shows F_1 scores on the same dataset with the default setting of 16 distance and 32 angle bins, but for varying support radii. There is a clear rise in performance from a too small (indiscriminative) radius to a clear peak around 50 mm, followed by smooth decay for too large (unstable) radii, where occlusions and cluttering elements start to impact performance.

Finally, Tab. 1 shows the performance of our descriptor using different matching methods, sorted from best to worst. For all tested metrics, we tried both the nearest neighbor distance and the nearest to second-nearest distance ratio [15] (also used in the previous section) and achieved best performances using the ratio for all metrics, except for the histogram intersection metric $\cap$. The L_2 and χ^2 metrics are defined in 2–3. The remaining metrics are defined as follows:

$$d_{L_1}(H_\mathcal{O}, H_\mathcal{S}) = \sum_{i=1}^{N} |H_{\mathcal{O},i} - H_{\mathcal{S},i}| \tag{5}$$

$$d_{\cap}(H_\mathcal{O}, H_\mathcal{S}) = \sum_{i=1}^{N} \min(H_{\mathcal{O},i}, H_{\mathcal{S},i}) \tag{6}$$

$$d_{KL}(H_\mathcal{O}, H_\mathcal{S}) = \sum_{i=1}^{N} \left(H_{\mathcal{O},i} \log \frac{H_{\mathcal{O},i}}{H_{\mathcal{S},i}} + H_{\mathcal{S},i} \log \frac{H_{\mathcal{S},i}}{H_{\mathcal{O},i}} \right) \tag{7}$$

$$d_{L_\infty}(H_\mathcal{O}, H_\mathcal{S}) = \max_{i=1}^{N} |H_{\mathcal{O},i} - H_{\mathcal{S},i}| \tag{8}$$

where KL denotes the symmetric Kullback-Leibler divergence.

	1000 RANSAC iterations				100 RANSAC iterations			
	UWA	Queen's	Spacetime	Kinect	UWA	Queen's	Spacetime	Kinect
ECSAD	0.907	0.731	0.769	0.773	0.681	0.402	0.606	0.585
FPFH	0.981	0.426	0.800	0.433	0.882	0.113	0.703	0.231
NDHist	0.941	0.437	0.667	0.426	0.800	0.165	0.562	0.351
RoPS	0.811	0.675	0.829	0.667	0.523	0.362	0.703	0.585
SHOT	0.888	0.779	**0.933**	0.825	0.662	0.410	**0.884**	0.757
SI	0.881	0.600	0.857	0.627	0.525	0.191	0.737	0.441
USC	0.738	0.429	0.769	0.854	0.574	0.248	0.625	0.789
PPFHist	**0.995**	**0.920**	**0.933**	**0.907**	**0.981**	**0.858**	**0.884**	**0.825**

Table 2: Recognition performances (F_1 scores) for the four real datasets. Best results are highlighted in bold. The cells with gray color are the four cases where our descriptor does not provide superior performance using only 100 RANSAC iterations, while still using 1000 iterations for the competing descriptor.

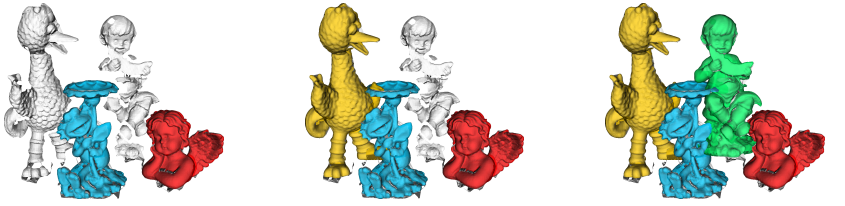
4.3 Recognition

Finally, we used our and all competing descriptors for 3D object recognition and 6 DoF pose estimation on the four real datasets. We used RANSAC [3, 8] with the correspondences given by the different descriptors as inputs. We then searched for the objects by sampling three correspondences at a time to generate candidate poses, which were then verified by their consensus set size. The parameters of RANSAC were tuned jointly on all four datasets and all eight descriptors to maximize the number of true positives and minimize the number of false positives to allow for best performances in terms of F_1 scores. The recognition results of this experiment are reported in Tab. 2 and a qualitative result using the three top performing descriptors on the Queen's dataset is shown in Fig. 5.

We used both a high and a low number of RANSAC iterations (1000 and 100, respectively). The 1000 iterations should be sufficient for producing at least one all-inlier correspondence triplet, thus providing optimistic performance numbers for all descriptors, since much of the noise from the matching process (*i.e.* the outlier correspondences) will be handled by RANSAC. To get a better picture of the dependency of the recognition performance on the strength of the matches, we also performed the same test with 100 RANSAC iterations. While many descriptor performances drop significantly, our descriptor maintains a high performance. Our descriptor outperforms all other descriptors in three of the datasets. For the fourth dataset, Spacetime, the performance of our descriptor is tied with SHOT[3]. Furthermore, with 100 RANSAC iterations we still outperform the competing descriptors in all but four cases (see Tab. 2), even when they are used with 1000 RANSAC iterations.

Timing-wise, substantial gains are achieved by reducing the number of RANSAC iterations. With 1000 iterations, each object instance is searched for in 1–2 s, depending on the dataset. Reducing the number of iterations linearly reduces this recognition time, thus allowing for objects to be recognized in a few hundred milliseconds with 100 iterations. For this specific dataset, the prior stages of descriptor estimation and matching for a full scene take a few seconds for our descriptor, making these stages the dominating processes when using 100 RANSAC iterations. These processes are significantly slower for many other descriptors.

[3]The SHOT descriptor has been optimized partially on the Spacetime dataset.



(a) ECSAD. (b) SHOT. (c) PPFHist.

Figure 5: Recognition output for the three top performing descriptors for the Queen's dataset, in this example for the scene named *im21*. The scene mesh is shown with gray colors and object alignments are shown with unique colors. The minimally required number of inliers for RANSAC is conservatively set to 5 % and only PPFHist is able to produce good enough correspondences to recognize all objects correctly when using 100 RANSAC iterations.

5 Conclusions and future work

This work contributed a new fast, yet highly discriminative local 3D descriptor for point cloud data. The descriptor uses a 2D histogram of position and normal information in a local neighborhood, and the χ^2 distance is used for matching these new descriptors.

We have presented a sensitivity analysis that allows us to choose the parameters in the descriptor creation process in a structured way. Furthermore, we have shown that our descriptor provides better matching results compared to seven state of the art descriptors on eight datasets (four artificial and four real world). Finally, we also presented experiments that show the applicability of the descriptor to 3D pose estimation where we match or surpass state of the art performances on the four real world datasets. Overall, we argue that our new descriptor is working better than previous descriptors and it does so robustly for different geometric structures.

The presented descriptor is only able to describe geometric structures. The integration of appearance information is a task that should be addressed in the future to make the descriptor richer and applicable to further use-cases. In the same stride, an automatic adjustment of parameters (feature radius is likely the most important parameter here) would be important to integrate. The results in this work have shown a very strong performance for multiple different datasets, all with the same parameter settings, but we foresee that this is not a given with all kinds of data. We have also deferred any comparisons against more recent learned descriptors, provided by *e.g.* deep learning methods. This is clearly an important next assessment to be done, although it requires other benchmark datasets that provide training data for the learning-based methods. Finally, we are interested in applying the new descriptor to different applications, such as object reconstruction, scene registration and face recognition.

Acknowledgments

The research leading to these results has been funded in part by Innovation Fund Denmark as a part of the project "MADE — Platform for Future Production" and by the EU FoF Project ReconCell (project number 680431).

References

[1] Prabin Bariya, John Novatnack, Gabriel Schwartz, and Ko Nishino. 3d geometric scale variability in range images: Features and descriptors. *International Journal of Computer Vision*, 99(2):232–255, 2012.

[2] Tolga Birdal and Slobodan Ilic. Point pair features based object detection and pose estimation revisited. In *IEEE International Conference on 3D Vision*, pages 527–535, 2015.

[3] Anders G Buch, Henrik G Petersen, and Norbert Krüger. Local shape feature fusion for improved matching, pose estimation and 3d object recognition. *SpringerPlus*, 5(1): 1, 2016.

[4] Hui Chen and Bir Bhanu. 3d free-form object recognition in range images using local surface patches. *Pattern Recognition Letters*, 28(10):1252–1262, 2007.

[5] Chin Seng Chua and Ray Jarvis. Point signatures: A new representation for 3d object recognition. *International Journal of Computer Vision*, 25(1):63–85, 1997.

[6] Bertram Drost and Slobodan Ilic. 3d object detection and localization using multimodal point pair features. In *Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, pages 9–16, 2012.

[7] Bertram Drost, Markus Ulrich, Nassir Navab, and Slobodan Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[8] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[9] Andrea Frome, Daniel Huber, Ravi Kolluri, Thomas Bülow, and Jitendra Malik. Recognizing objects in range data using regional point descriptors. In *European Conference on Computer Vision*, pages 224–237, 2004.

[10] Yulan Guo, Ferdous Sohel, Mohammed Bennamoun, Min Lu, and Jianwei Wan. Rotational projection statistics for 3d local surface description and object recognition. *International Journal of Computer Vision*, 105(1):63–86, 2013.

[11] Yulan Guo, Mohammed Bennamoun, Ferdous Sohel, Min Lu, Jianwei Wan, and Ngai Ming Kwok. A comprehensive performance evaluation of 3d local feature descriptors. *International Journal of Computer Vision*, 116(1):66–89, 2016.

[12] Stefan Hinterstoisser, Vincent Lepetit, Naresh Rajkumar, and Kurt Konolige. Going further with point pair features. In *European Conference on Computer Vision*, pages 834–848, 2016.

[13] Andrew E. Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):433–449, 1999.

[14] Troels Bo Jørgensen, Anders Glent Buch, and Dirk Kraft. Geometric edge description and classification in point cloud data with application to 3d object recognition. In *International Conference on Computer Vision Theory and Applications*, 2015.

[15] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[16] Ajmal S Mian, Mohammed Bennamoun, and Robyn Owens. Three-dimensional model-based object recognition and segmentation in cluttered scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1584–1601, 2006.

[17] Iordanis Mpiperis, Sotiris Malassiotis, and Michael G Strintzis. 3-d face recognition with the geodesic polar representation. *IEEE Transactions on Information Forensics and Security*, 2(3):537–547, 2007.

[18] Marius Muja and David G Lowe. Scalable nearest neighbor algorithms for high dimensional data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11): 2227–2240, 2014.

[19] Radu Bogdan Rusu and Steve Cousins. 3d is here: Point cloud library (pcl). In *IEEE International Conference on Robotics and Automation*, pages 1–4, 2011.

[20] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *IEEE International Conference on Robotics and Automation*, pages 3212–3217, 2009.

[21] Samuele Salti, Federico Tombari, and Luigi Di Stefano. Shot: unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding*, 125:251–264, 2014.

[22] Fridtjof Stein and Gérard Medioni. Structural indexing: Efficient 3-d object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):125–145, 1992.

[23] Babak Taati and Michael Greenspan. Local shape descriptor selection for object recognition in range data. *Computer Vision and Image Understanding*, 115(5):681–694, 2011.

[24] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique shape context for 3d data description. In *Proceedings of the ACM workshop on 3D object retrieval*, pages 57–62, 2010.

[25] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Performance evaluation of 3d keypoint detectors. *International Journal of Computer Vision*, 102(1-3):198–220, 2013.

[26] Eric Wahl, Ulrich Hillenbrand, and Gerd Hirzinger. Surflet-pair-relation histograms: a statistical 3d-shape representation for rapid classification. In *Fourth International Conference on 3-D Digital Imaging and Modeling.*, pages 474–481, 2003.

[27] Andrei Zaharescu, Edmond Boyer, and Radu Horaud. Keypoints and local descriptors of scalar functions on 2d manifolds. *International Journal of Computer Vision*, 100(1): 78–98, 2012.