Contents lists available at ScienceDirect

Computer Vision and Image Understanding

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

A performance evaluation of point pair features

Lilita Kiforenko*[a], Bertram Drost[b], Federico Tombari[c], Norbert Krüger[a], Anders Glent Buch[a]

[a] The Maersk Mc-Kinney Moller Institute, University of Southern Denmark, Odense, Denmark
[b] MVTec Software GmbH, München, Germany
[c] Technische Universität München, Garching b. München, Germany

ARTICLE INFO

Keywords:
PPF
Point pair features
Object detection
Object recognition
Pose estimation
Feature description

MSC:
41A05
41A10
65D05
65D17

ABSTRACT

More than a decade ago, the point pair features (PPFs) were introduced, showing a great potential for 3D object detection and pose estimation under very different conditions. Many modifications have been made to the original PPF, in each case showing varying degrees of improvement for specific datasets. However, to the best of our knowledge, no comprehensive evaluation of these features has been made. In this work, we evaluate PPFs on a large set of 3D scenes. We not only compare PPFs to local point cloud descriptors, but also investigate the internal variations of PPFs (different types of relations between two points). Our comparison is made on 7 publicly available datasets, showing variations on a number of parameters, e.g. acquisition technique, the number of objects/scenes and the amount of occlusion and clutter. We evaluate feature performance both at a point-wise object-scene correspondence level and for overall object detection and pose estimation in a RANSAC pipeline. Additionally, we also present object detection and pose estimation results for the original, voting based, PPF algorithm. Our results show that in general PPF is the top performer, however, there are datasets, which have low resolution data, where local histogram features show a higher performance than PPFs. We also found that PPFs compared to most local histogram features degrade faster under disturbances such as occlusion and clutter, however, PPFs still remain more descriptive on an absolute scale. The main contribution of this paper is a detailed analysis of PPFs, which highlights under which conditions PPFs perform particularly well as well as its main weaknesses.

1. Introduction

Through the last three decades, many different 3D feature descriptors have been proposed. Usually, they are divided into two categories: global feature based methods (which describe the object using one global feature, e.g. Siddiqi et al., 1998 and Wahl et al., 2003) and local feature based methods (which describe the object using point neighbourhoods, e.g. Guo et al., 2014 and Wu et al., 2010). In the field of 3D object pose estimation, local feature descriptors have become more popular than global ones, since the local nature of such features makes the description tolerant to occlusions and clutter. Global descriptors are used primarily for object shape matching (object retrieval). The global descriptors represent the full object by some structure, e.g. skeletal graphs (Siddiqi et al., 1998) or a histogram over some relational features (Wahl et al., 2003). The global descriptors can be computationally expensive and require segmentation and full object shape, which makes them less stable under high occlusion. On the other hand, most local descriptors are by themselves computationally less expensive and more robust towards clutter and occlusion, but instead incur additional computation time in the following stages of matching and hypothesis verification. There have been proposals of combining both global and local descriptors. For example in Wu et al. (2010), a manifold harmonic analysis is used to design an isometry-invariant descriptor for 3D object shape comparison. Another type of a descriptor that captures local and global information is point pair features (PPFs) (Drost et al., 2010). This point relational descriptor has shown very successful object detection on different 3D datasets.

Local histogram based feature descriptors have been investigated in many works (Buch et al., 2016; Guo et al., 2016), mostly in order to design a better, more discriminative, faster and stable descriptor. Also, some work has been done in exhaustive evaluation of local descriptors, where popular descriptors have been evaluated on many datasets in order to conclude which ones are more robust, descriptive, scalable and efficient (Guo et al., 2016). In contrast to local descriptors, such evaluation has not been done yet for the PPF descriptor.

In this paper, we present a comprehensive analysis of the PPF descriptor and provide a comparison with several 3D local histogram feature descriptors (SHOT, ECSAD, FPFH, USC, SI). Here the 3D local

* Corresponding author.
E-mail address: lilita@kiforenko.eu (L. Kiforenko).

http://dx.doi.org/10.1016/j.cviu.2017.09.004
Received 21 November 2016; Received in revised form 12 June 2017; Accepted 17 September 2017
Available online 20 September 2017
1077-3142/

histogram feature descriptors serve as a baseline for the comparison to the PPF descriptor. Feature evaluations and comparisons are performed on 7 publicly available datasets, which contain large varieties of objects and scenes recorded by different sensors and for various purposes. One of the data sets (Sølund et al., 2016) provides 3204 scenes and 45 models and exceeds other data sets in terms of size and complexity. We start our investigations with a systematic evaluation of the relational space in the PPF descriptor (Section 5). We evaluate 19 different PPF variations and determine the best PPF for different data sources. Our results show that on average, the original PPF performs best, but there are special cases, e.g. where adding colour to the PPF descriptor boosts the performance.

After the initial PPF analysis, we perform a systematic comparison between the PPFs and 5 local histogram feature descriptors (Section 6). The results depend on the dataset, for example, if the data is recorded by a high precision laser scanner, then PPFs significantly outperform local histogram descriptors. The opposite occurs when the input data is less detailed and noisy, such as data from Kinect-like devices, where the highest performance is achieved by local histogram features.

Moreover, we investigate robustness towards noise, occlusion and clutter (Section 6.3). Our results show that performance of PPFs is decreasing significantly faster under increasing occlusion, clutter and noise compared to most local histogram descriptors.

One of the used datasets divides the objects into three different categories, which allows us to show feature performance for different object categories (Section 6.2). Our results also show that PPFs in comparison to local histogram features have a significant drop of initial precision (i.e. the inlier rate of the top ranked matches), which indicates that PPFs need to be used in more robust object detection algorithm.

After feature performance evaluations on a point-wise object-scene correspondence level, we present object detection and pose estimation results (Section 7). In order to be consistent, we compute object poses using the same RANSAC pose estimation pipeline for every used feature. We also provide object detection and pose estimation results for voting based pose clustering for PPFs as a reference.

Our object detection and pose estimation results show that PPFs clearly outperform local histogram features for two datasets, containing scenes with low quality and high levels of occlusion. On the other hand, if the scenes contain high-quality reconstructions with moderate occlusions and clutter, our results show that local features perform best. In one case, that is for a Kinect-based dataset, we get similar performances for local features and PPFs. Interestingly, this dataset does indeed differ from the others as it has low occlusions, but overall the quality of the point cloud data is poor.

In the end of the work, we provide a discussion of the achieved results (Section 8).

2. Related work

Object detection and pose estimation from 3D data have been a popular topic for many researchers, which lead to a development of a rich variety of feature descriptors. This section presents an overview of the popular feature descriptors, but mostly focusing on point pair features and their modifications.

Before the 3D sensors were available, a lot of work was focused on designing feature descriptors for images (2D data). The designed popular descriptor are still used to this data, for example, SIFT , SURF etc. Due to the descriptors good performance, some of them were extended to the 3D data. For example, 3D-SURF (Knopp et al., 2010) or SI-SIFT (Bayramoglu and Alatan, 2010).

The SURF or speeded up robust feature was initially inspired by SIFT and became a faster and more robust feature descriptor. The descriptor computes the Haar wavelet response of the feature point neighbourhood, it is scale and rotation invariant and can be also used as interest point detector.

The SI-SIFT feature descriptor integrates shape index with the SIFT 2D feature descriptor. It has shown a good performance for the data which is rotated, scaled and occluded. Shape index is a value, which describes the principal curvature of the 3D point of interest. For each 3D point, the shape index is computed and used to build a 2D image, which represents the depth discontinuities. SIFT features are computed using build 2D image as input.

Another often used feature descriptor is Normal Aligned Radial Feature (NARF) (Steder et al., 2010), which is not only a feature descriptor but also a robust interest point extractor. NARF extracts interest points, which have a stable normal and a significant change in depth.

Sun et al. (2009) proposed a Heat Kernel Signature (HKS) descriptor, which describes the heat distribution of the feature point neighbourhood. They showed that this descriptor is stable towards scale change and is isometric invariant. HKS capture both the local and global properties of the feature point.

PPFs were introduced for object recognition by Drost et al. (2010). Their point pair features use quite primitive relationships between any two points, such as distance and angle between normals. Together with a hash table and an efficient voting scheme, the method performs well in the case of occlusion, clutter and noise. The features were tested both on synthetic and real datasets. For a real dataset Drost et al. achieved a success rate of up to 97% for objects with occlusion levels less than 84%.

The method quickly became popular and many modifications have been proposed. Choi et al. (2012) proposed to use different types of relations for point pair features, for example, boundary to boundary relations or relations between two lines which are created by the edge points. Using these edge point relations decreases the number of features both for training and matching; consequently, it increases the detection speed. This modification shows in particular good performance for industrial (mostly planar) objects.

Kim and Medioni (2011) proposed to add a visibility context to the original PPF, creating a five-dimensional feature vector. They used three types of visibility - space, surface, invisible surface. Adding a visibility parameter improves the PPF matching. The approach was tested using a view-based object models on a data captured by RGB-D camera. The result shows clear outperformance of the original method on the same data.

Choi and Christensen (2012) described another modification of PPFs by adding a colour component to the traditional 4 dimensional point pair feature, creating CPPF - a 10 dimensional descriptor. The results showed good performance for 10 textured household objects in highly occluded and cluttered scenes.

Drost and Ilic (2012) computed PPFs for geometric edges (boundaries and silhouettes). In this case, the PPFs were computed slightly differently by the use of edge gradients. The evaluation was made on the ACCV3D dataset (Hinterstoisser et al., 2012), where the proposed method significantly improves the PPF descriptor in highly occluded scenes.

All the methods mentioned above were using very similar detection pipelines. Birdal and Ilic (2015) proposed another one, where a scene is first segmented and then PPFs are computed for each segment. The method was tested on the ACCV3D dataset (Hinterstoisser et al., 2012) with an average success rate of 88.77%, which is higher compared to original PPFs (81.18%).

The work by Tuzel et al. (2014) presents an approach for learning features. They showed that certain pairs do not have enough discriminative information (for example, pairs on the same planar area). Specifically, the features that were learned were the weights for the hash table bins and dummyTXdummy- weights for the object model points. The method was tested on two datasets. The results showed improvement in object detection for both of the tested datasets compared to the original PPF method.

The research into PPFs is still ongoing. Recently, a new method has been presented by Hinterstoisser et al. (2016), which proposes to use a different point sampling and pose voting approach. With their modification, PPFs become more robust towards noise and clutter.

As stated earlier, the PPF became popular and many modifications were developed. But with each modification came usually a new dataset and a new use case. To the best of our knowledge, there is no work that would systematically test different modifications of PPFs. In this paper, we investigate different variations of PPFs on the same datasets in order to see which feature components are the strongest. We also present results for local histogram feature descriptors that are computed for comparison. The work by Guo et al. (2016) presented a performance comparison of popular 3D local descriptors (not PPFs) on a sizable scale. We do not aim to repeat their work, but to use several local histogram feature descriptors as a baseline.

3. 3D Feature descriptors

This section provides a summary of all the features used in our comparisons. As stated earlier, in this work we evaluate PPFs and its variants against several local histogram-based 3D descriptors. All local histogram feature descriptors are well-known, therefore here they are presented only briefly, and more information can be found in related papers. We selected 5 popular and publicly available 3D local feature descriptors, which according to the research results in Buch et al. (2016) and Guo et al. (2016) have shown high performance over a wide range of datasets and therefore can be considered state of the art for local 3D object description. Guo et al. (2016) have shown that Rotational Projection Statistics (RoPS) can be used as robust feature descriptors. Unfortunately, RoPS require a triangular mesh and all used features are computed on point cloud data. Therefore we omitted RoPS in this work.

3.1. Single point features

3.1.1. SHOT or signature of histograms of orientation (Tombari et al., 2010b)

SHOT is a local histogram surface feature descriptor. For each single point, it computes a unique, repeatable local reference frame using an eigenvalue decomposition. Using the reference frame, it builds a spherical grid around the input point that divides supporting points into grid cells. At each grid sector, it computes a weighted cosine of the relative normal angle and bins the result into a local histogram for that cell. SHOT combines all local histograms into one descriptor of length 352. In the last stage, the descriptor is normalized to an Euclidean norm of 1.

3.1.2. SI or spin image (Johnson and Hebert, 1999)

SI is a 2D histogram descriptor. The histograms are obtained by spinning a 2D plane around the normal of a point and accumulating the number of 3D points falling on such plane. The plane is discretized using a 2D grid. For our tests, we used a 153 dimensional SI.

3.1.3. ECSAD or equivalent circumference surface angle descriptor (Jørgensen et al., 2015)

ECSAD is a surface-edge feature descriptor, which in the original work is also applied for the task of detecting edges in point clouds. The main focus of ECSAD is to represent the relative angles between opposite surface cells around an edge point. It uses a special azimuth binning to achieve this; the supporting sphere is split into 6 cells (60 degrees each), then it splits each of the 6 cells depending on its radial increment (one split per each increment). A local reference frame is also computed using the eigenvalue decomposition of the supporting points. Then, all support points are mapped to the corresponding spatial bins using radial and azimuth coordinates. For each of the bins, an average angle is computed, and interpolation is used to fill empty bins. These angles are used directly as the descriptor values. The recommended length of the descriptor is 30, which corresponds to 4 radial and 3 azimuth levels.

3.1.4. PFH/FPFH or (Fast) point feature histogram (Rusu et al., 2009)

PFH is multi-dimensional histogram descriptor that stores the information about the relations of the point pairs in the neighbourhood of the reference point. For each point pair, a coordinate frame is computed that allows to compute the angular difference between the three normals. Those values are binned into a histogram. The final descriptor is the concatenation of the three values. The PFH is a computationally expensive descriptor. FPFH is the fast version of the PFH. It removes the links between the reference point neighbours, but keeps only the pairs between the reference point and its neighbours. For each dimension, FPFH calculates a separate histogram and after combines the three histograms together. The final descriptor is computed by weighted merging each reference point histogram with its neighbour points histograms.

3.1.5. 3DSC/USC or (3D, Unique) shape context (Frome et al., 2004; Tombari et al., 2010a)

3DSC creates a sphere around the reference point, where the “north pole” of it is aligned with the point normal. The sphere is divided into regions with linearly spaced divisions across azimuth and elevation and logarithmically spaced across the radial dimension. The descriptor contains the weighted count of the points falling at each sphere regions. The 3DSC descriptor cannot deal with rotations, because the sphere “north pole” is aligned with the point normal, which locks the two axes and leaves the spheres azimuth “rotation free”. To overcome this, the point sphere is rotated around the point normal N times. N corresponds to a number of azimuth divisions. It also means that each point has a total of N descriptors. USC is the extension of the 3DSC descriptor that fixes the multiple descriptors per point problem by computing the local reference frame as presented in the SHOT feature.

3.2. Point pair features

The original PPF descriptor (Drost et al., 2010, Eq. (3.2)) is a 4 dimensional descriptor, which encodes the relations between two surface points p_1 and p_2.

$$PPF(p_1, p_2) = (d, \angle(n_1, p_2 - p_1), \angle(n_2, p_2 - p_1), \angle(n_1, n_2))$$

where d is the Euclidean distance between p_1 and p_2, n_1 and n_2 are the point normals and $\angle(n_1, p_2 - p_1)$ and $\angle(n_2, p_2 - p_1)$ are the angles between the point normals and the direction vector between the two points.

In our implementation of the PPF feature, all feature values are normalized to be within [0, 1]. For angles, this normalization is trivial, since these are already restricted to the bounded interval [⁻1, 1]. For normalizing the distance relations, we restrict these numbers to the interval of [0.005, object model diameter]; 0.005 is an experimentally selected minimum range. Note that in Section 5, we perform initial analyses where we investigate the performance of different relations between two points in order to show if the original 4-dimensional PPF is the best choice.

4. Experimental data

For our feature evaluations, we have selected 7 publicly available datasets, which we will describe in detail below. The datasets were selected based on different criteria, for example, quality of the data, acquisition technique, object model completeness (view based or full 3D), colour availability, etc. The features that we use in our tests are

Table 1
Used datasets for feature evaluation.

Name	Acquisition	Objects	Scenes	#Models	#Scenes	Data points	Colour	Avg. occlusion/clutter(%)
UWA	Minolta vivid laser scanner	3D	2.5D	6	50	188	No	75.47/67.07
Retrieval	Synthetic	3D	3D	6	18	18	No	0/0
Random views	Synthetic	3D	2.5D	6	36	324	No	59.59/66.51
Queens	Minolta vivid laser scanner	3D	2.5D	5	63	208	No	69.57/68.08
Kinect	Microsoft kinect	2.5D	2.5D	28	17	49	Yes	NA
Space time	Spacetime stereo	2.5D	2.5D	6	12	24	Yes	NA
DTU	Structured light scanner	3D	2.5D/3D	45	3204	4059	No	89.58/89.80

computed for a single object in one scene at a time. Each object-scene pair we name a *data point*. We decided to use 7 datasets, because our goal is to make a comprehensive feature evaluation, which requires a large variety of data sources. The selected datasets are collected using several sensors and contain various object and scene types. The UWA[1] (Mian et al., 2006), Random views,[2] Retrieval,[2] Queens[3] (Taati et al., 2007) and DTU (Sølund et al., 2016) datasets all contain full object models. The UWA dataset provides scenes without background and much clutter (i.e. data not belonging to the objects). The Queens dataset is similar to UWA, but it has some background data and clutter. The Retrieval and Random views datasets provide synthetic scenes with three different noise levels. The Kinect[2] and Space time[2] datasets provide view based object models. The two datasets also provide colour information. Most of the datasets have a very limited number of scenes. Therefore, in our evaluation we use a new dataset (DTU) which contains 3204 scenes with a high variety of objects captured by a structured light scanner. A summary of the datasets is shown in Table 1.

4.0.1. UWA dataset

The UWA dataset contains 5 full object models and 50 view based scenes (Fig. 1). The data is captured with Minolta Vivid 910 laser scanner with a resolution of 640 × 480. The scenes were generated by randomly placing 4 or 5 of the objects together. The dataset provides objects and scenes as high resolution meshes. For each object in a scene, the ground truth 4 × 4 transformation matrix is provided together with the amount of occlusion and clutter. Some of the provided ground truth poses contain errors, they were manually corrected for experiments in this paper. One of the objects, *rhino*, is usually excluded from evaluation due to large holes during the scan procedure. For our evaluation, we also excluded the *rhino* object. The number of data points for the UWA dataset that is used for evaluation is 188 (50 for the object *chef*, 48 for the object *chicken*, 45 for the object *parasaurolophus* and 45 for the object *T-rex*).

4.0.2. Retrieval and random views datasets

The Retrieval and Random views datasets are synthetic (Fig. 2). They were built by using the models from the Stanford 3D scanning repository.[4] Both datasets provide each scene view with 3 different noise levels - 0.01%, 0.03% and 0.05% of the resolution. The difference between the datasets is that the Retrieval dataset contains only one object in a scene (the scene is the object with different noise levels), while the Random views dataset contains synthetic views of a combination of three random objects. Both datasets are using the same 6 objects. The number of data points for Retrieval is 18 (6 scenes × 3 noise levels) and 324 for Random views (36 scenes × 3 noise levels × 6 models). For each object in a scene, the ground truth pose is provided. Both datasets provide object and scene models as high resolution meshes.

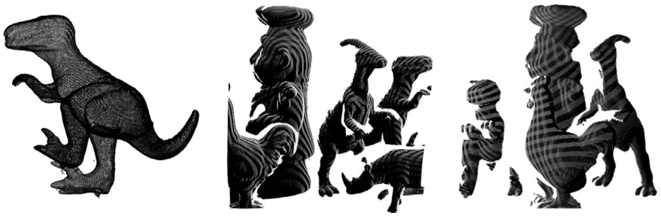
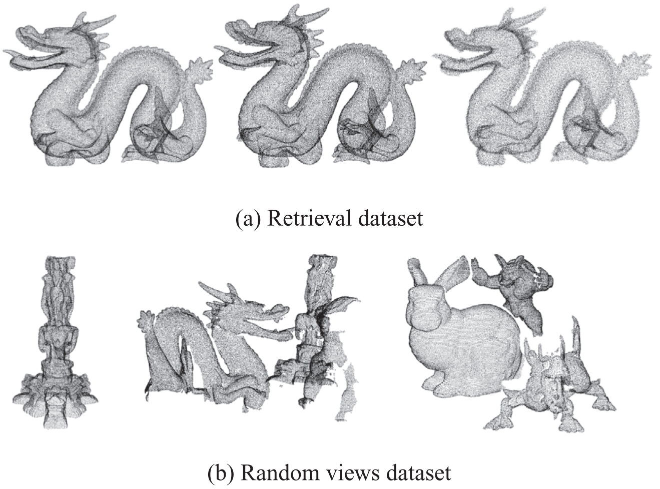


Fig. 1. UWA dataset example - one random object model and two scenes.



(a) Retrieval dataset

(b) Random views dataset

Fig. 2. Used datasets examples - one random object model and two scenes.

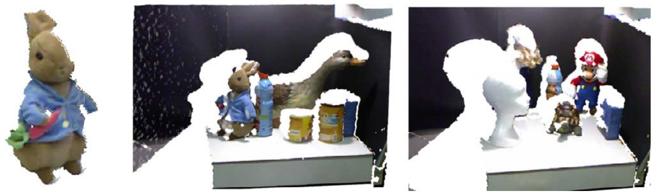


Fig. 3. Kinect dataset example - one random object model and two scenes.

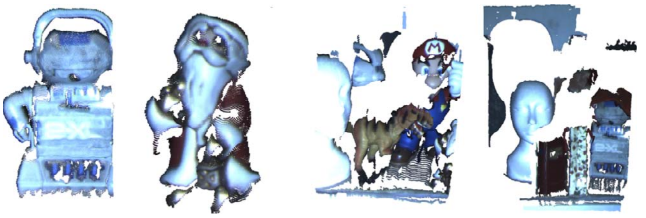


Fig. 4. Space time dataset example - two random object models and scenes.

4.0.3. Kinect dataset

The Kinect dataset (Fig. 3) was recorded using the *Microsoft Kinect v1* sensor. It contains 17 scenes and 28 models. The 28 models represent

[1] http://staffhome.ecm.uwa.edu.au/~00053650/recognition.html.

[2] http://vision.deis.unibo.it/keypoints3d/.

[3] http://rcvlab.ece.queensu.ca/~qridb/.

[4] https://graphics.stanford.edu/data/3Dscanrep.

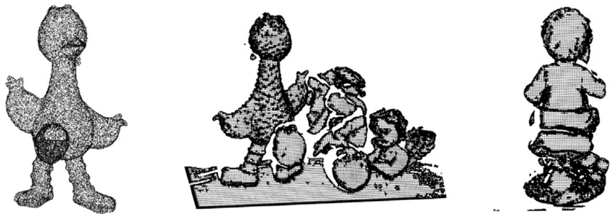


Fig. 5. Queens dataset example - one random object model and two scenes.

6 different objects, which are captured from various views. Each scene comes with a ground truth pose for only a specific object view. Models and scenes are given with colour information. The number of data points for the Kinect dataset is 49.

4.0.4. Space time dataset

The Space time dataset (Fig. 4) was recorded using the Spacetime stereo acquisition technique. The Spacetime algorithm reconstructs the point depth using a combination of two different techniques for the correspondence matching - the traditional feature search in left and right camera images in the spatial domain and feature search across multiple frames in the time domain. The final depth is obtained using triangulation (Davis et al., 2005).

The dataset contains 6 view based object models and 12 scenes. For each scene, the dataset provides the ground truth poses of two object views. The number of data points for the Space time dataset is 24 (12 scenes × 2 objects). Models and scenes are given with colour information.

4.0.5. Queens dataset

The Queens dataset (Fig. 5) is similar to the UWA dataset described above. The Queens dataset contains data acquired by the Minolta vivid scanner. The data contain five full object models and 63 scenes. Compared to the UWA dataset, the Queens dataset has a larger variety of objects present in a scene, but the quality of the scenes for the Queens dataset is lower. The number of data points for the Queens dataset is 208.

4.0.6. DTU dataset

The DTU is a recently introduced dataset (Fig. 6), which consists of 45 objects and 3204 scenes. The dataset is recorded systematically in an environment without ambient light and with controlled illumination. The system includes an industrial robot in a dark chamber and a high precision structured light scanner. The 45 object models are recorded with a high precision structured light scanner and a rotation table. The recorded object views are merged in order to get a full 3D object model. The scenes are recorded from 11 different view points. Each scene contains 10 objects. In this work, we used the dataset in the initial state, in a future more ground truth poses should be added to the dataset. Some of the objects are highly occluded. We removed all data points, where the occlusion for the object is more than 98%. The number of data points after the removal of high occluded scenes is 4059.

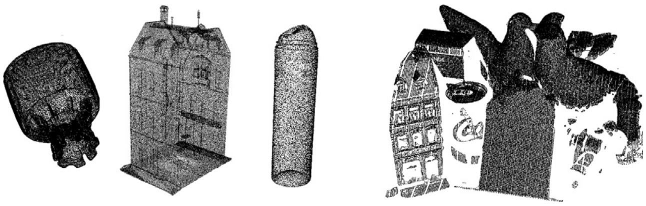


Fig. 6. DTU dataset example - three random object models and one scene.

5. Initial PPF analysis

As stated in previous sections, many PPF variations have been proposed recently (Choi and Christensen, 2012; Choi et al., 2013). Typically they are compared to the original PPF descriptor, but the comparisons are carried out on a relatively small or inaccessible dataset. In this section, we perform initial experiments in order to analyse the different variations of the PPF descriptor. The analyses are made at the feature correspondence level in order to isolate the underlying feature performance and not bias our results on any subsequent object recognition algorithm. A similar approach is also used in Buch et al. (2016) and Guo et al. (2016). In the end of the section, we determine the most optimal PPF descriptor(s) for different data sources.

5.1. Evaluation protocol

In the literature, different evaluation protocols have been proposed. We adopt the evaluation protocol similar to the one proposed by Tombari et al. (2010b), where a set of feature points is randomly extracted from the model along with the closest corresponding points in the scene. The experimental protocol outlined in this section is specifically designed for the purpose of performing an in-depth analysis of the different PPF variations. In the next section (Section 6), we will use a slightly different protocol to be able to provide a fair comparison between PPFs and local descriptors for matching tasks relevant to more realistic 3D object recognition scenarios.

Input data: The input data for our tests in this experiment is 500 points on the object surface and 500 corresponding scene points. The 500 points were selected in the following manner: we take a random object point, transform it into the scene and find the closest point in the scene within a small radius; if such a point is found, we add the corresponding object and scene points to our input data; the procedure is repeated until we find 500 corresponding points. 500 points are selected, because it is enough to cover the whole surface and it is not too computationally exhaustive for the second-order feature computations required by PPFs. An example of 500 selected random points is shown in Fig. 7.

Feature computation: First, we compute all possible pairs for the 500 points and then compute the different PPF variations for each pair. We discard all pairs, where the Euclidean distance between two points is smaller than a predefined *min*, because too close points will not provide discriminative information. The selected *min* value is 0.005 m.

Feature matching: After computing different PPFs for the object and corresponding scene point pairs, for each object PPF we find the closest match in the scene PPFs using a k-d tree with the Euclidean distance metric. Using the ground truth pose information, we evaluate how many of the closest matches between the object and scene PPFs are correct.

5.2. Result of initial analysis

In our implementation of the point pair relations, we were influenced by the work of Drost et al. (2010), Drost and Ilic (2012), Choi et al. (2013) and Choi and Christensen (2012).

Point pair relation combinations. The original PPF contains four

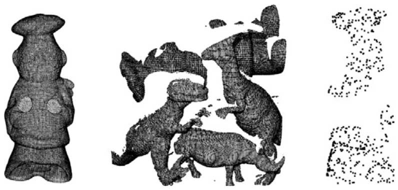


Fig. 7. Randomly selected points on the object surface that are not occluded in the scene. Left - downsampled object point cloud, middle - downsampled scene point cloud, right - 500 random selected points on the object.

Table 2
PPF relation variations. *d* is the Euclidean distance between two points, *n* is points normalm *rgb, hsv* are points colour components.

No	Relations	No	Relations
1	(d)	11	(d_1, n_1d, n_2d)
2	(n_1d)	12	(d_1, n_1d, n_1n_2)
3	(n_2d)	13	(d_1, n_2d, n_1n_2)
4	(n_1n_2)	14	(n_1d, n_2d, n_1n_2)
5	(d, n_1d)	15	$(d_1, n_1d, n_2d, n_1n_2)$
6	(d, n_2d)	16	$(d_1, n_1d, n_2d, n_1n_2, rgb)$
7	(d, n_1n_2)	17	$(d_1, n_1d, n_2d, n_1n_2, hsv)$
8	(n_1d, n_2d)	18	$(d_1, n_1d, n_2d, n_1n_2, rgbdiff)$
9	(n_1d, n_1n_2)	19	$(d_1, n_1d, n_2d, n_1n_2, hsvdiff)$
10	(n_2d, n_1n_2)		

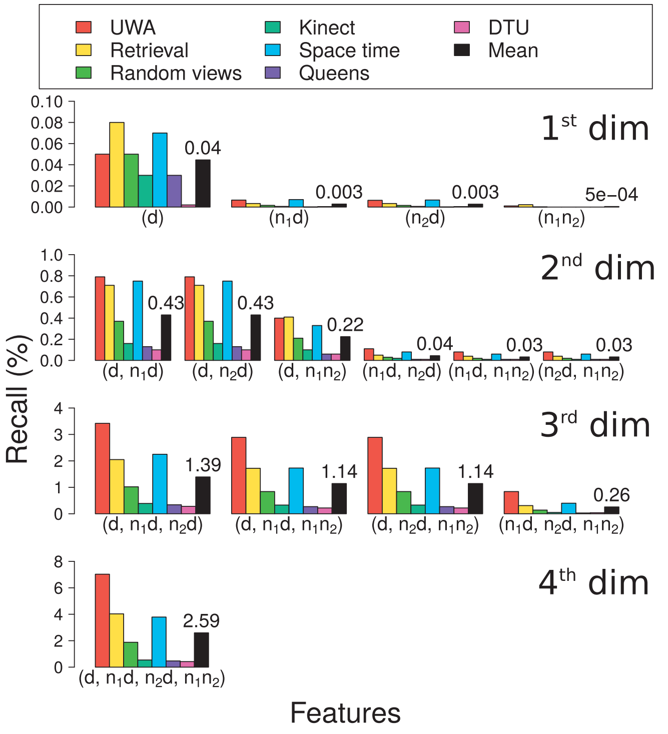


Fig. 8. Point pair relation combination result.

relations between two points. The question arises - *is it necessary to have all four relations?* To the best of our knowledge, there is no research which could answer this question. Therefore we provide results for all possible combinations of these relations. The combination variations are presented in Table 2 (1–15).

A more visual overview of these relations is also shown in Fig. 10. Fig. 8 presents the average performance of different combinations of the PPF relations for each of the 7 datasets. The results show that the one dimensional features are performing poorly compared to the four dimensional feature. The most discriminative relation between the two points is the distance. For example, the performance of the three-dimensional feature, which contains the relations between the point normals - (n_1d, n_2d, n_1n_2) is on average performing worse then the two dimensional features, which contain the distance component. All in all, we can conclude (as expected) that the four dimensional PPF $(d_1, n_1d, n_2d, n_1n_2)$ is the best performing feature across all eight datasets.

Colour. Choi and Christensen (2012) proposed to add a colour component to the PPF, creating a 10 dimensional feature vector. Their results show a significant object detection rate increase compared to the original (geometric) PPF. To encode the colour information, they used the HSV colour space.

In our work, we test the performance of the colour PPF feature. We explore the performance of the RGB and HSV colour spaces on two datasets - Kinect and Space time, which contain colour information. We implemented two different ways of encoding the colour information:

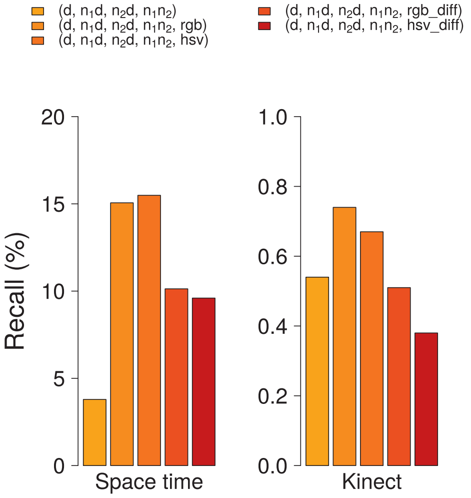


Fig. 9. Point pairs colour result for Space time and Kinect datasets.

- the raw colour values of the two PPF endpoints in RGB or HSV colour space (i.e. a 6 dimensional vector), and
- the absolute difference between the two endpoint colour values in RGB or HSV space (a 3 dimensional vector)

The tested combinations are shown in Table 2 (16–19) and Fig. 10. We used the previously computed 4-dimensional PPF $(d_1, n_1d, n_2d, n_1n_2)$ values as a baseline.

The results for PPF with colour are shown in Fig. 9. For both datasets, there is an increase in performance when using colours. For the Space time dataset, there is a significant absolute increase (≈ 10%) when adding colour information to the PPF. There is no significant difference between colour spaces, but it is better to use raw point colour values instead of the difference vector. For the Space time and Kinect datasets, the HSV colour space is performing slightly better than RGB. Choi and Christensen (2012) proposed to have a different weight for the V channel in HSV colour space. We experimented with varying weights for the V channel, but it did not significantly change the performance on our datasets.

We can conclude from the results and visual observations that if the colour information of the scene is consistent with the colour values of the object model, then using the colour will clearly boost the feature performance. But when there is different light exposures between object model and scene (for example, for such objects as *head* and *robot* from

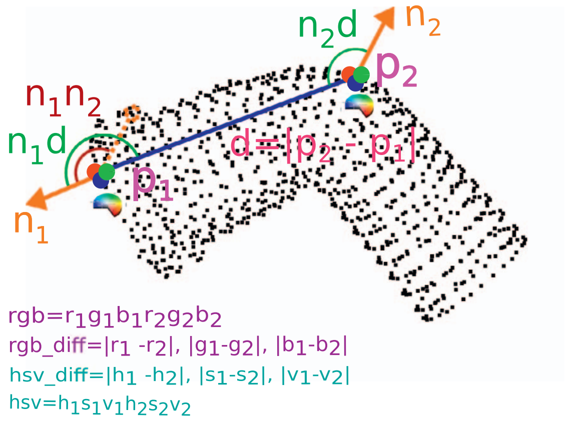

n2d
n2
p2
n1n2
n1d
d=|p2 - p1|
p1
n1
rgb=r1g1b1r2g2b2
rgb_diff=|r1 -r2|, |g1-g2|, |b1-b2|
hsv_diff=|h1 -h2|, |s1-s2|, |v1-v2|
hsv=h1s1v1h2s2v2

Fig. 10. Relations used in the PPF descriptor.

Space time dataset (Fig. 4)), then adding the colour component will decrease the feature performance.

5.3. Conclusion on initial analyses

The results in this section have shown that the strongest PPF feature is 4 dimensional - (d_1, n_1d, n_2d, n_1n_2) for all datasets, except the Kinect and Space time datasets. For the Kinect and Space time datasets it is better to use the colour version of the PPF descriptor - (d_1, n_1d, n_2d, n_1n_2, rgb) or (d_1, n_1d, n_2d, n_1n_2, hsv).

For the rest of our work we will use the 4 dimensional (d_1, n_1d, n_2d, n_1n_2) PPF descriptor for all datasets and the 10 dimensional (d_1, n_1d, n_2d, n_1n_2, hsv) PPF descriptor for the Space time and Kinect datasets.

As stated earlier, the results show that the distance between two points is more discriminative than the angle between point normals. We believe that the reason for the distance to be more robust is because it does not change much with noise. The distance is mostly dependable on the sensor resolution/quality, which is slightly worse for RGB-D cameras and better for laser scanners. However, normals represent a vector which is perpendicular to the surface and they are computed by fitting a plane. One of the most important parts is to determine the size of the surface the plane will be fit to. In this paper, we used constant surface size for whole object/scene, which can lead to incorrect normals at some object parts. Additionally, it is known that sensor quality degrades with increasing distance, which means that normals become even more unstable when the object is far away and reconstruction is noisier.

The colour experiment result shows that using absolute colours is better than the colour differences. We believe that the absolute colours perform better because the colour differences do not represent the actual colours, which was beneficial for the datasets we used. Working with colour faces the difficult problem of non-linear illumination effects that can dramatically change the colour values of the same object under different viewpoints/sensors. However, when illumination conditions are rather constant, the absolute colours can provide valuable information. In the future work, we would like to experiment with the gradients of gray values, which should be more robust towards illumination changes.

6. Comparative experiments between PPF and local histogram feature descriptors

In the previous section, we have treated a PPF feature descriptor as a single feature vector. For example, for m object points (throughout the experiments m has been set to 500) we got in the order of m^2 feature vectors, then using the k-d tree we found the closest match between the $\sim m^2$ object feature vectors and the m^2 scene feature vectors and computed how many matches are correct. Even though such a comparison protocol can be used to compare the same type of features, it has some issues:

- it does not take into account points, which belong to other objects and the background,
- it is hard to compare PPFs to other features: local histogram based descriptors produce one feature vector per point, while PPF will compute many feature vectors per point, which results in an unfair comparison

Additionally, individual PPFs are low dimensional (4–10 dimensions), while local histogram features encode much more information (for example the SHOT descriptor length is 352). Therefore, in practice, voting schemes are used for PPFs (Birdal and Ilic, 2015; Choi and Christensen, 2012; Drost et al., 2010) and in our work, we propose to use similar voting scheme while comparing PPFs with local histogram features.

In order to make the comparison between PPF and local histogram features, we define a protocol, which includes the voting method for PPFs. This is detailed in the section below. The results are shown as Precision-Recall (PR) curves (Eqs. (1) and (2)), which is a common way to display feature correspondence performance (Guo et al., 2016). The precision will show the number of correct matches in a whole scene, while recall will show the number of correct matches out of the possible correct matches (not all points in the scene will belong to the object).

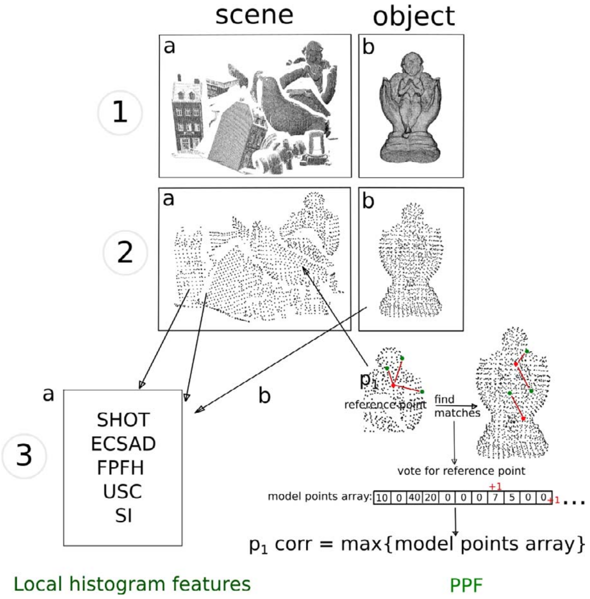


Fig. 11. Feature comparison protocol. 1ab - original scene and object model point cloud, 2ab - downsampled scene and object model.

$$Precision = \frac{Count\ of\ correct\ matches}{Total\ count\ of\ matches} \tag{1}$$

$$Recall = \frac{Count\ of\ correct\ matches}{Count\ of\ possible\ correct\ matches} \tag{2}$$

We also compute the area under the PR curve (named AUC) for each feature descriptor. The AUC shows a single aggregated value, which measures the feature performance over the whole PR space.

6.1. Comparison protocol

The overall process of the protocol used in this section is shown in Fig. 11. First, we downsample the scene (1a) and the object model (1b) using a voxel grid with a leaf size of 0.01 m (2ab).

Local histogram features: For each downsampled scene and object point, we compute the 5 tested local histogram feature descriptors (3a). Then for each feature point in the scene, we find a matching object feature point using the nearest neighbour distance ratio technique (Lowe, 2004).

PPFs: For each downsampled object model, we compute all possible PPFs within the *min* distance, as it is explained in Section 5.1. We also eliminate pairs, where the distance between two points is bigger than a predefined *max*, which is simply set to the object model diameter, because we are interested only in finding one object at a time. The same constraints are used by Drost et al. (2010). We can consider this as a *training* stage. Then we perform matching with voting.[5] For each downsampled point in the scene (in the following called a *reference point*), we create a voting array with one entry per object model point (3b). For each reference point, we compute all possible PPFs within the *min* and *max* distance. Then we match these scene PPFs with the stored object model PPFs, i.e. for each reference point PPF, we find the closest

[5] The voting procedure outlined here is a similar method for matching PPFs as used by Drost and Ilic (2012), and has been devised after communications with the authors.

Table 3
The tuned parameters for local histogram features. The values are shown in meters.

Dataset	Feature descriptor				
	SHOT	ECSAD	FPFH	USC	SI
UWA	0.03	0.03	0.02	0.03	0.04
Retrieval	0.1	0.05	0.04	0.05	0.1
Random views	0.02	0.02	0.02	0.04	0.08
Kinect	0.06	0.06	0.06	0.1	0.1
Space time	0.04	0.04	0.05	0.06	0.08
Queens	0.04	0.04	0.01	0.04	0.09
DTU	0.04	0.02	0.02	0.02	0.08

PPFs from the object model using a k-d tree with a radius search in the feature space of the PPF. Then we perform voting, where each matching pair votes for the reference point. We assume that the scene reference point lays on the object, which means that the object model point with the maximum number of votes should correspond to the scene reference point.

The voting for PPFs allows us for each object model point to have one corresponding match in the scene as it is the case for the local histogram feature matching. By comparing to the ground truth, we can find the number of correct matches and build the PR curves.

6.2. Results

All results are computed using the protocol described in Section 6.1. For the PR curves in the PPF case, we are using the normalised number of votes each reference point receives. The normalisation is performed by dividing the votes by the number of point pairs for each of the reference points.

All feature descriptors parameters (e.g. support radii) were tuned for each dataset using three random object-scene pairs and are shown in Table 3. The feature matching results for all our datasets are shown in Figs. 12 and 13 and analysed in the next subsections. An AUC summary is shown in Table 4.

6.2.1. UWA dataset

Fig. 12(a) shows the PR curves and AUC values for the tested feature descriptors for the UWA dataset. The best performing feature for the UWA dataset is PPF with a significant recall difference compared to the other features. Then follows FPFH, SHOT, ECSAD and SI features. The lowest achieved performance is by USC. The ranking of features for this dataset is consistent with the work of Guo et al. (2016).

From the result, we can conclude that PPFs performs better than local histogram features for high quality laser scanner data with rich shape variation, no background noise, but also with a high degree of occlusion.

6.2.2. Retrieval and random views dataset

Fig. 12(b) and (c) shows the PR curves and AUC values for 6 different feature descriptors for the Retrieval and Random views dataset. We used only the scenes from the dataset with the lowest amount of noise as in Guo et al. (2016).

The best performing features for Retrieval dataset are SHOT and USC, then follows ECSAD, FPFH, SI and PPF. For Random views dataset, the best performing feature is PPF, then follows FPFH and SHOT with similar performance. Lower performance is shown by SI, ECSAD and USC. The ranking for local histogram features is almost consistent with the work of Guo et al. (2016). For some features, we achieve higher AUC values for Retrieval dataset and lower for Random views dataset, compared to the results reported by Guo et al. For example, the SHOT features AUC in our work for Retrieval dataset is 0.932, while Guo at. al for the same dataset reports 0.544 and for the Random views dataset we get lower absolute values.

PPFs for Retrieval dataset achieves the lowest performance and for Random views the highest compared to local histogram features. One explanation could be that Retrieval dataset scenes are full 3D object models and Random views dataset scenes are synthetically rendered 2.5D views. For full 3D scenes, there are point pairs on the front and back side of the object. Ideally, those point pair normals are directed in opposite direction, creating relative normal angles in $\angle(n_1, n_2)$ around 180 degrees. It is possible that this creates a lot of similar feature vectors, which leads to a degradation in performance.

Even though Retrieval and Random views datasets share the same object models, the overall performance of the Random views dataset is lower, which is also expected due to a more complicated scenes and the presence of occlusion and clutter.

6.2.3. Kinect dataset

The Kinect dataset contains wall background data, which we removed in order to speed up computations. Fig. 12(d) shows the PR curves and AUC values for 7 different feature descriptors for the Kinect dataset. The precision values for this dataset are lower than for the UWA, Retrieval and Random views datasets. One of the explanations could be the low quality of the data, which is recorded with the Kinect device. Also, object models are only 2.5D, which is not a perfect representation of the object.

The highest performance for this dataset is achieved by USC, then with a significant gap follows the SHOT feature. The next in rank features are SI, FPFH, ECSAD and both PPF features. The high performance of USC indicates that it is quite robust towards the noise. Our later investigations in Section 6.3.1 back up this hypothesis.

Both PPF versions perform poorly for this dataset. From our investigation in Section 5, the colour version of PPF should outperform the original PPF. And the results show that the PPF colour has a higher precision compared to the PPF, but the overall number of correct correspondences is lower. The higher recall value for PPF shows that there are correct matches between the object and the scene, but the ranking of the votes seems to be failing, which means that highest votes are assigned to more wrong matches. This could be explained by repeated structures in a scene.

6.2.4. Spacetime dataset

Fig. 12(e) shows the PR curves and AUC values for the tested feature descriptors for the Spacetime dataset. The USC and SHOT descriptors significantly outperform the others, which is again consistent with the findings by Guo et al. (2016). The SI and FPFH show average performance, then follows PPF colour. ECSAD and PPF perform the worst. The PPF colour significantly outperforms the original PPF, which is consistent with our findings in Section 5.2.

6.2.5. Queens dataset

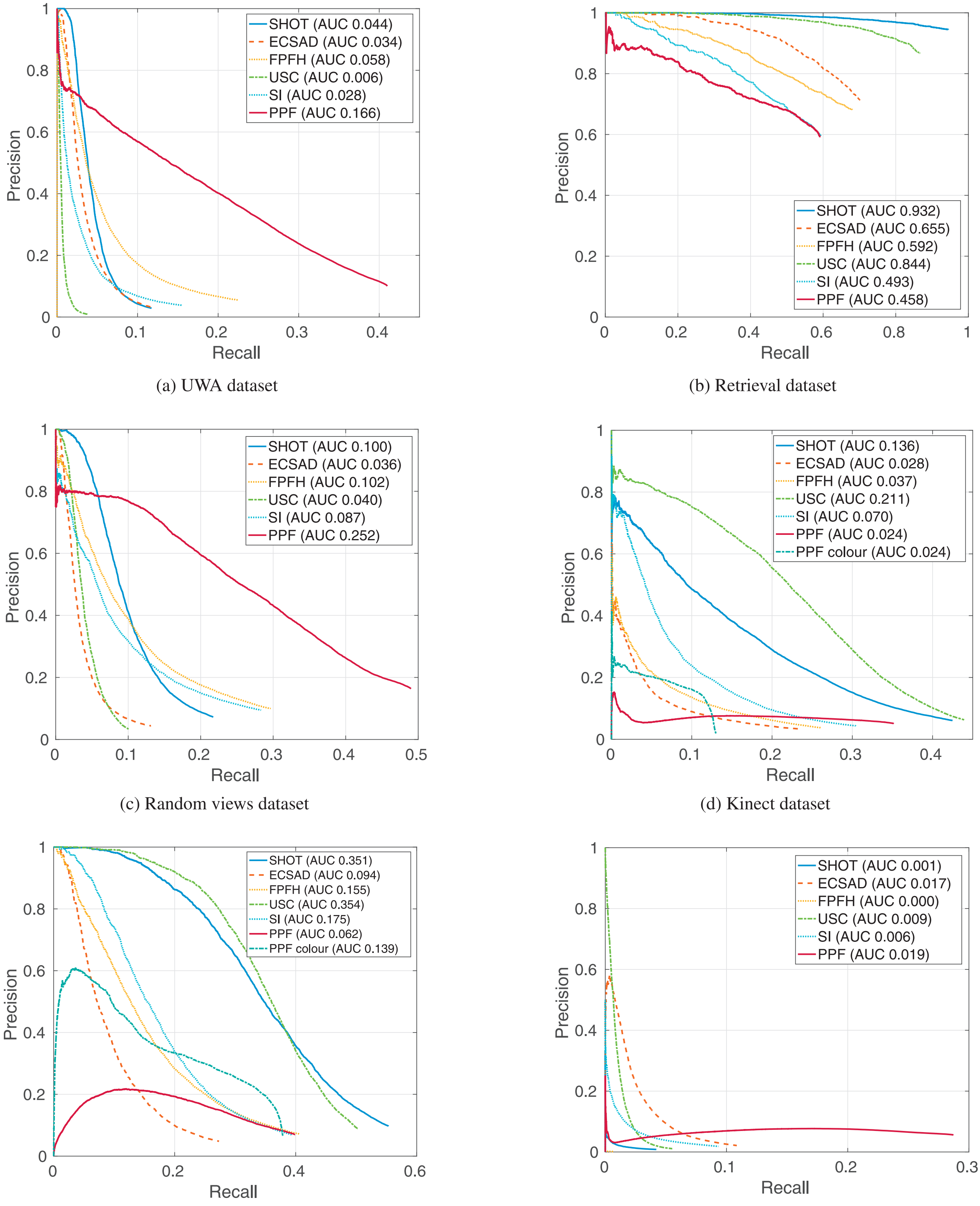
Fig. 12(f) shows the PR curves and AUC values for the Queens dataset. The best performing features are PPF and ECSAD, then follows USC and SI. Very low performance are obtained by SHOT and FPFH on this dataset. The overall feature performance for the Queens dataset is low. We believe this is due to the quality of the data.

Similar to the Kinect dataset, the PPF vote ranking is not performing as expected. A lot of high votes are assigned to the background points, which leads to incorrect matches. The overall recall value for the PPF is significantly higher than the recall values for the local histogram features, which indicates that there is a large number of correct correspondences.

6.2.6. DTU dataset

Fig. 13 shows the PR curves and AUC values for the DTU dataset. This dataset has a large number of data points (4059). Therefore we also provide PR curves for three different object types - *Geometric complex objects, Cylindrical objects* and *Flat or box shaped objects*.

The AUC values (Fig. 13(a)) for DTU are the lowest compared to



(a) UWA dataset

(b) Retrieval dataset

(c) Random views dataset

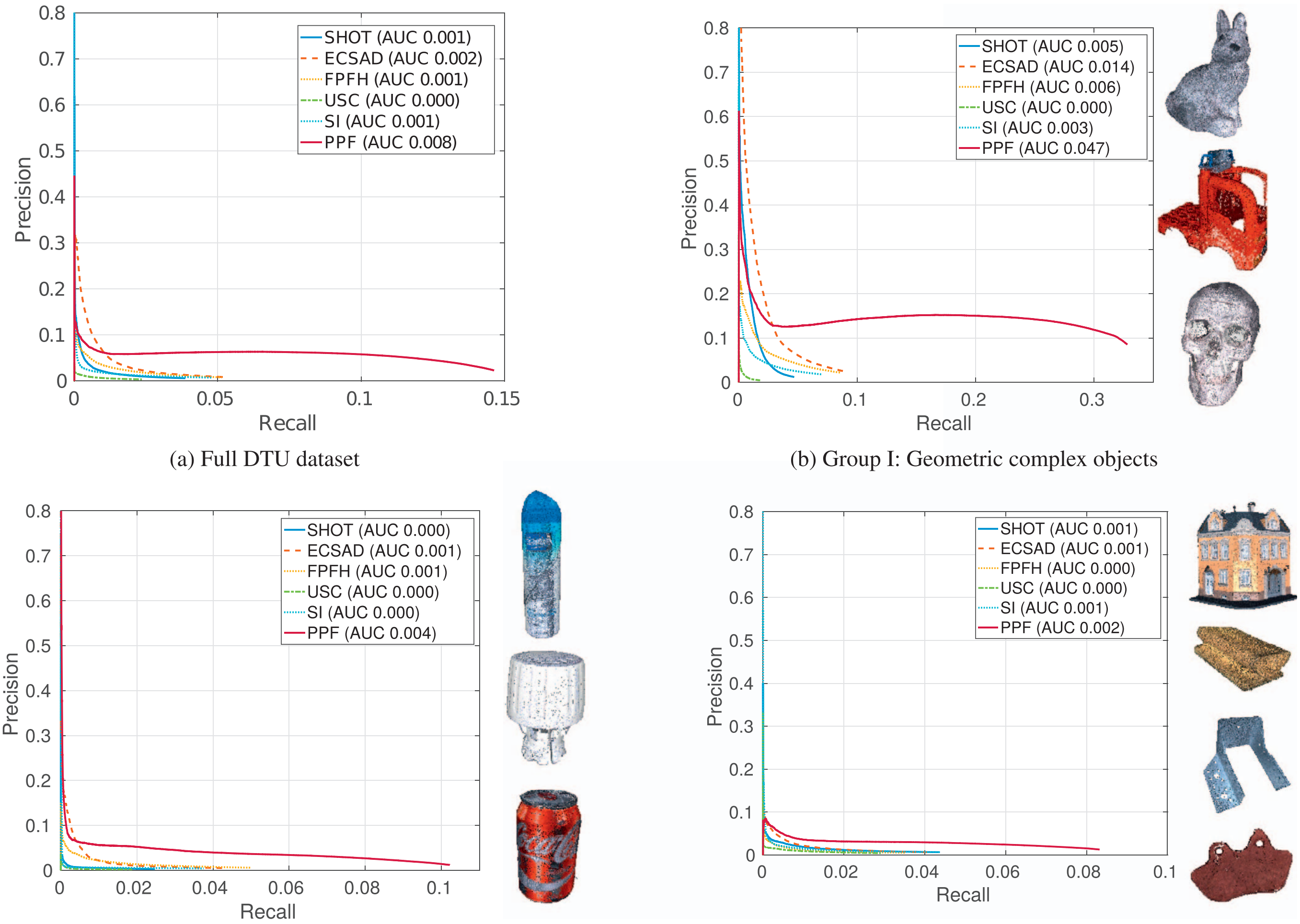
(d) Kinect dataset

(e) Spacetime dataset

(f) Queens dataset

Fig. 12. Feature performance results.

other datasets. There is, however, a clear difference in feature performance between Group I (Fig. 13(b)) and Group II/III (Fig. 13(c)/(d)). This is also expected, because cylindrical and flat objects have a lot of repeated local structures, and local histogram features are mostly made to detect objects with rich shape variations (Kiforenko et al., 2015). Please note that the mean occlusion level for this dataset is 89.58% and clutter - 89.80%. The high occlusion and clutter levels is the reason for very low feature performance. Some examples of highly occluded objects are shown in Fig. 14. All graphs for DTU dataset show that PPFs clearly outperform local histogram features in recall value.



(a) Full DTU dataset

(b) Group I: Geometric complex objects

(c) Group II: Cylindrical objects

(d) Group III: Flat or box shaped objects

Fig. 13. Feature performance results for DTU dataset.

6.2.7. Discussion and conclusion on results

All subfigures in Figs. 12 and 13 show that there is a problem using the PPF voting scheme for matching (all graphs starts with low precision), because in many cases the highest votes correspond to wrong matches. Based on this we can conclude that we cannot rely on e.g. the 10%–20% (or some other low value) best matches for object detection algorithms for PPFs, as it is usually done for local histogram features (Buch et al., 2016). Instead, an accurate matching and detection algorithm needs to reason with all correspondences. For an object detection algorithm, the solution would be to use multiple voting peaks as it is done in e.g. Drost and Ilic, 2012. The evaluation of how this affects the performance could be done in a future work.

In conclusion, according to Table 4, the most frequent top performer over the datasets is PPF. This feature achieved the highest scores for 4 out of 7 datasets, then follows SHOT (2/7) and USC (2/7). The highest mean AUC value across all datasets is achieved by the SHOT feature (0.224), then follows USC (0.209) and with some gap PPF (0.141). The easiest dataset according to the mean AUC values is Retrieval (0.662), on a second place is Space time (0.199) and third is the Random views dataset (0.103). The hardest dataset for the presented features is DTU

Table 4
The summary of AUC values for all 7 datasets. The blue highlights show the best performing descriptor for each dataset. For two datasets, Space time and Queens, the top two performing descriptors are highlighted, because the performance differences are insignificant. In the bottom row, we highlight the two top performers over all datasets with bold. Similarly, the two datasets for which all descriptors perform best are highlighted with bold in the rightmost column.

Dataset	Feature descriptor							Mean
	SHOT	ECSAD	FPFH	USC	SI	PPF	PPF colour	
UWA	0.044	0.034	0.058	0.006	0.028	0.166	-	0.056
Retrieval	0.932	0.655	0.592	0.844	0.493	0.458	-	**0.662**
Random views	0.100	0.036	0.102	0.040	0.087	0.252	-	0.103
Kinect	0.136	0.028	0.037	0.211	0.070	0.024	0.024	0.084
Space time	0.351	0.094	0.155	0.354	0.175	0.062	0.139	**0.199**
Queens	0.001	0.017	0.000	0.009	0.006	0.019	-	0.009
DTU	0.001	0.002	0.001	0.000	0.001	0.008	-	0.002
Mean	**0.224**	0.124	0.135	**0.209**	0.123	0.141	0.082	-

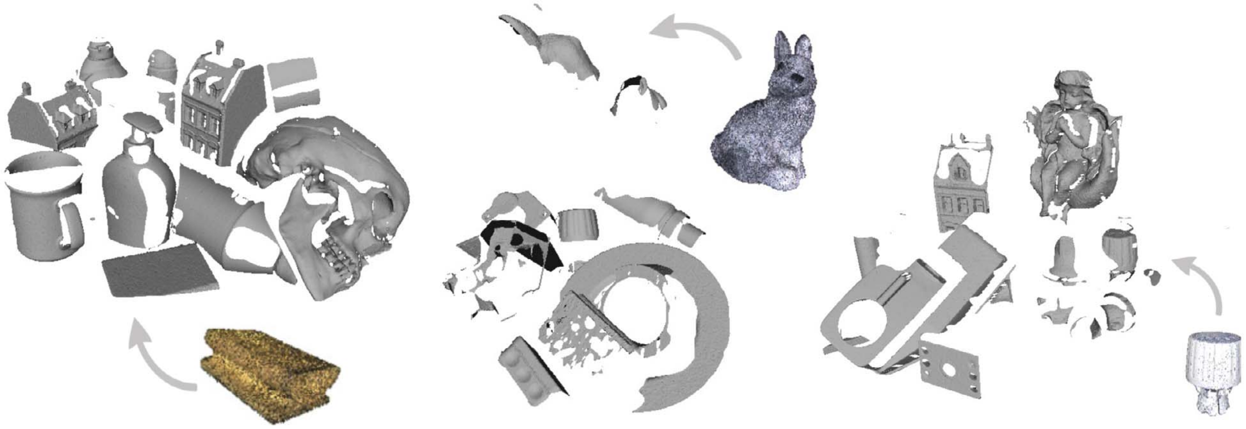


Fig. 14. Some examples why DTU dataset is challenging for the features.

with a mean AUC of 0.002, which sets a new baseline to develop even better feature descriptors in the future. High occlusion and clutter is a challenge for every used feature, perhaps methods that take a prior context knowledge into account would perform more successfully for this dataset.

6.3. Further sensitivity analyses

In this section, we analyse the effects of various sources of error for the matching process. In particular, we test the robustness of each feature under increasing noise levels and under varying levels of missing data (occlusions) and irrelevant data (clutter). For all our analysis, we use the same protocol as in the section above (Section 6.1). The results are displayed as AUC value graphs.

6.3.1. Noise

The Retrieval and Random views datasets provide scenes with three different noise levels. Fig. 15 displays the AUC values at these noise levels for the two datasets combined.

Based on relative numbers, the results show that the most stable feature against noise is USC. Then follows SI and ECSAD, PPF and SHOT. The most unstable is FPFH. On average, the best performing feature is PPF, which significantly outperforms local histogram features. Note that the provided AUC value is the combined value for both datasets.

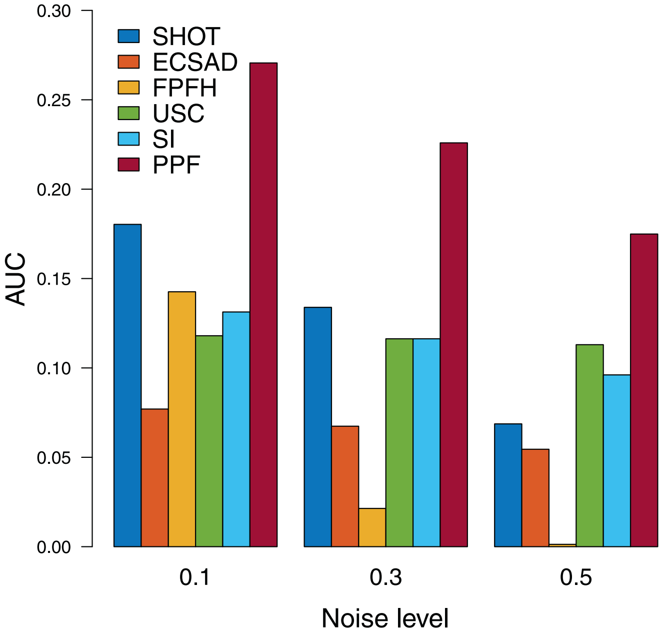


Fig. 15. Feature performance in the presence of noise (Retrieval and Random views datasets).

6.4. Occlusion and clutter

For the datasets with full 3D object models, we can compute how much of the object is occluded in a scene and how much clutter the scene contains. This allows us to evaluate the feature descriptor robustness towards occlusion and clutter. Some of the datasets already provide the amount of occlusion and clutter (e.g. UWA), but in order to be consistent, we recomputed the amount of occlusion and clutter for each dataset. Our occlusion and clutter values are very similar to the provided ones. The formula for computing occlusion is shown in Eq. (3) and for computing clutter is shown in Eq. (4). The same formulas are used in several earlier works, e.g. Guo et al., 2016 and Drost et al., 2010. We chose do exclude the Retrieval and Random views datasets, because these datasets consist of synthetically generated scenes.

$$Occlusion = 1 - \frac{Count\ of\ object\ points\ in\ the\ scene}{Total\ count\ of\ object\ points} \tag{3}$$

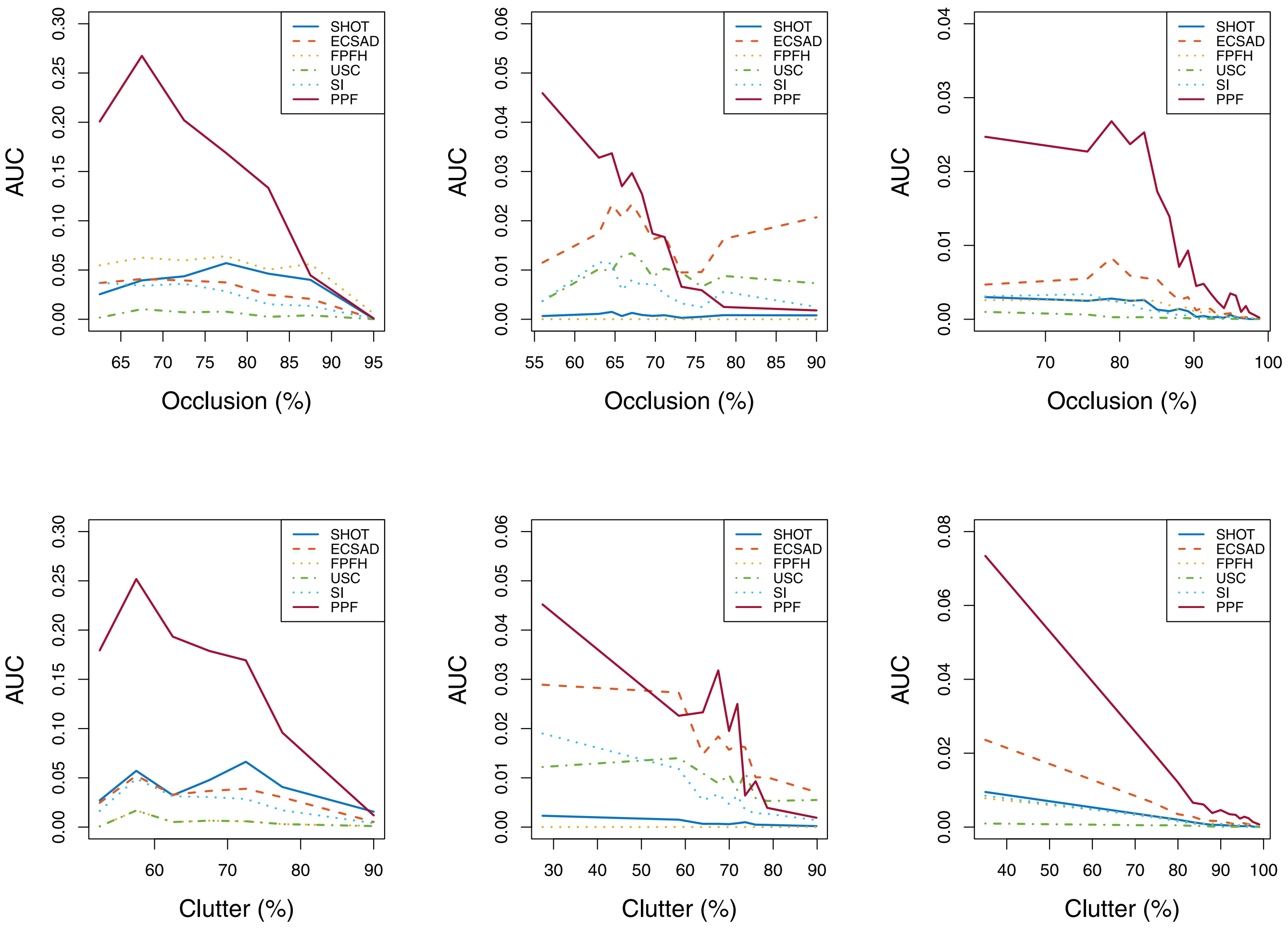
$$Clutter = 1 - \frac{Count\ of\ object\ points\ in\ the\ scene}{Count\ of\ points\ in\ the\ scene} \tag{4}$$

Fig. 16 displays the AUC values for three datasets for different occlusion and clutter levels. The results show that the PPF is more unstable under occlusion and clutter compared to the local histogram features. On average the local histogram feature performance does not change rapidly with the increase of the occlusion and clutter, which is due to the small support radii. Even though the PPF is more unstable than local histogram features, for two out of three datasets the overall performance of the PPF is higher even in highly occluded and cluttered scenes.

7. Object detection and pose estimation

Feature correspondences are showing the strength of the features, and these correspondences can be provided as inputs to a subsequent detection algorithm, which uses a robust estimation method for finding the object pose. Different methods can be used to get the object pose (e.g. voting, RANSAC). In this work, after computing the scene-object correspondences (as described in Section 6.1), we use a Correspondence Rejector Sample Consensus algorithm to remove wrong correspondences and get the object pose. After, the pose is refined with the ICP algorithm (Besl and McKay, 1992). We used a PCL[6] version of

[6] http://www.pointclouds.org.



(a) UWA dataset

(b) Queens dataset

(c) DTU dataset

Fig. 16. Feature performance under occlusion and clutter.

Correspondence Rejector Sample Consensus and ICP.

There are different ways to determine if the estimated and refined output pose is correct. One way is to compute the difference between the ground truth and the found pose. Unfortunately, for some of the datasets, the object models coordinate frame is not placed in a centre, which can lead to significant errors between the ground truth and found poses, even if the found pose produces an alignment which is just slightly off. Therefore we decided to use the method proposed by Hinterstoisser et al. (2012). We transform the object model using the ground truth pose and the found pose and compute the L_2 distance between the transformed object points (Eq. (5)).

$$\textit{alignment error} = \frac{1}{N}\sum_{i=1}^{N} \|p_{gt} - p_{found}\|_2 \tag{5}$$

where p is a 3D point vector.

Ideally, the computed alignment error should be extremely small, but due to inaccuracies of the ground truth pose, we allow for higher threshold values. The threshold values were found empirically and they differ for different dataset due to ground truth pose and data quality.

We provide object detection and pose estimation results for the 7 features we used in Section 6.1. For all 7 features, the final pose is obtained using a RANSAC object detection pipeline with ICP. Originally, PPFs are used with a voting based pose clustering and not in a RANSAC-like approach. The voting based pose clustering is possible, because by having two points we can reliably compute an object pose. Unfortunately, such an approach is not possible for local histogram features, which are based on single point matches. Therefore, in order to compare PPFs and local histogram features, we decided to use the same object detection method. However, in order to see the difference between PPF performance using different methods, we also present PPF result using the original pose clustering approach (we denote them as PPF$_{voting}$). PPF$_{voting}$ is based on the original PPF implementation (Drost et al., 2010), which is now a part of the commercial product HALCON (ver. 12) by MVTec.[7] Please note that PPF$_{voting}$ results are presented as a baseline and to show the potential of PPFs in a different object detection pipeline, but they cannot be used in direct comparison to the performance of local histogram features.

The object detection and pose estimation results are displayed in two tables. Table 5 reports object detection results, which are simply the relative amount of poses that produce high-quality alignments according to Eq. (5). Table 6 shows the alignment error means and standard deviations as well as medians for each feature for only the correct detections.

The results show that PPF$_{voting}$ outperforms our feature implementations both in smaller alignment error and higher recognition rates for all datasets. The smaller alignment errors are obtained using a coarse-to-fine ICP implementation.

In the following text, we focus on evaluating the performance of the 5 local histogram features and two our implemented PPFs. Most of the results are as expected from the previous displayed precision-recall results (Figs. 12 and 13). For the UWA dataset, our implementation of

[7] http://www.mvtec.com.

Table 5
Object detection rate (%). Please note that PPF$_{voting}$ uses a different pose aquisition technique and is shown as a reference.

Feature	Dataset							Mean	DTU object types		
	UWA	Retrieval	Random views	Kinect	Space time	Queens	DTU		I	II	III
SHOT	51.06	100	95.37	93.88	100	29.33	6.43	68.01	19.89	2.76	7.47
ECSAD	64.36	100	76.85	87.76	87.50	67.31	10.54	70.62	45.30	9.76	7.09
FPFH	84.57	100	98.15	93.88	91.67	1.44	8.92	68.38	48.43	8.47	3.08
USC	9.57	100	54.63	93.88	95.83	37.02	2.61	56.22	4.79	0.74	4.11
SI	75.00	100	93.52	85.71	100	61.54	7.39	74.74	30.94	4.05	6.01
PPF	88.30	100	94.44	93.88	95.83	80.77	19.36	81.80	63.90	30.20	16.07
PPF$_{colour}$	-	-	-	67.35	75.00	-	-	71.18	-	-	-
PPF$_{voting}$	93.62	100	99.07	100	100	99.52	39.59	90.26	77.35	89.69	43.61
Mean	66.64	100	87.43	89.54	93.23	53.85	13.55	-	41.51	20.81	12.49

Table 6
Pose alignment results (mm) for correctly detected objects.

Feature	Dataset													
	UWA		Retrieval		Random views		Kinect		Space time		Queens		DTU	
	$\mu \pm \sigma$	m	$\mu \pm \sigma$	m	$\mu \pm \sigma$	m	$\mu \pm \sigma$	m	$\mu \pm \sigma$	m	$\mu \pm \sigma$	m	$\mu \pm \sigma$	m
SHOT	1.7 ± 0.8	1.5	0.4 ± 0.1	0.4	1.6 ± 0.9	1.4	5.7 ± 6.6	3.1	1.4 ± 1.5	1.0	6.2 ± 8.7	4.0	2.1 ± 1.6	1.6
ECSAD	1.7 ± 0.7	1.5	0.4 ± 0.1	0.4	1.5 ± 0.7	1.4	5.8 ± 7.0	3.1	1.5 ± 1.6	0.9	4.9 ± 5.1	3.9	2.1 ± 1.4	1.6
FPFH	1.7 ± 0.7	1.5	0.4 ± 0.1	0.4	1.6 ± 0.8	1.4	7.4 ± 10.4	3.0	1.5 ± 1.6	1.0	N/A	N/A	2.1 ± 1.5	1.6
USC	2.0 ± 0.9	2.0	0.4 ± 0.1	0.4	1.6 ± 0.8	1.5	5.8 ± 6.8	3.1	1.4 ± 1.6	0.9	5.1 ± 5.1	4.2	1.8 ± 1.1	1.5
SI	1.6 ± 0.7	1.4	0.4 ± 0.1	0.4	1.6 ± 0.8	1.4	6.1 ± 7.3	2.8	1.5 ± 1.5	1.0	5.4 ± 4.0	6.6	1.9 ± 1.4	1.5
PPF	1.1 ± 0.3	1.0	0.4 ± 0.1	0.4	1.6 ± 0.9	1.4	5.7 ± 6.7	2.9	1.5 ± 1.8	0.8	4.4 ± 2.4	3.9	2.5 ± 1.8	1.9
PPF$_{colour}$	–	–	–	–	–	–	7.8 ± 8.5	5.4	1.3 ± 1.8	0.8	–	–	–	–
PPF$_{voting}$	0.07 ± 0.04	0.06	0.02 ± 0.001	0.02	0.05 ± 0.08	0.04	5.4 ± 6.7	2.6	1.6 ± 2.0	0.7	4.6 ± 2.4	4.2	2.5 ± 1.9	2.0

PPF outperforms local histogram features and produces smaller pose errors. The Retrieval dataset is not challenging for any feature, therefore all objects are detected correctly with a very small alignment error.

For the Random views dataset, the FPFH feature produces the highest object recognition rate, which is not consistent with the PR curves, where PPFs are significantly outperforming all local histogram features. The performance difference between two features is small, which could be explained by the randomness of RANSAC algorithm. More interesting is that the AUC values for this dataset show that USC feature is slightly better than ECSAD, but recognition results show an absolute difference of 20% between those features, where ECSAD outperforms USC. It seems that the higher recall helps ECSAD to outperform USC, although the latter has a higher precision and AUC.

Based on the PR results, we predicted that the performance of PPF$_{colour}$ for the Kinect and Space time datasets should be better than PPF, but our detection results show the opposite. For the Kinect dataset, it could be explained by a much higher recall value for PPF. The objects that are not correctly recognised for Space time dataset for the PPF$_{colour}$ feature are the *robot* and *head*. Those objects have a bright surface, which “makes” objects uniformly white. Also, there are white parts of background present in some scenes, which causes the PPF$_{colour}$ feature to produce a lot of false positives.

Compared to other datasets, the object detection for the Queens and DTU datasets is lower, which is consistent with the PR results. The high occlusion level for the DTU dataset makes it the hardest dataset for object detection.

Some of the object detection results are shown in Fig. 17. Overall datasets, the PPF feature has the highest recognition rate, and USC the lowest. The recognition rate for PPF and PPF$_{voting}$ for some datasets differ significantly and for some the difference is small.

8. Conclusion

In this work, we have evaluated the performance of point pair features (PPFs). We started with presenting an initial analysis, where we investigated the space of possible relations between two points for building PPFs. The conclusion of the analysis was that the original 4-dimensional PPF is generally the best choice, unless consistent object-scene colour information is available, in which case a 10-dimensional PPF containing additional colour information (PPF$_{colour}$) shows a better performance. Next, we presented the main part of our work - a performance comparison between PPFs and 5 local histogram features (SHOT, ECSAD, FPFH, USC, SI) on 7 publicly available datasets. The results were presented as precision-recall (PR) curves of object-scene point correspondences. The performance of features was dependent on the dataset considered. For high resolution data, PPF outperformed other features. For noisier data, SHOT and USC showed good performance. We also performed an additional sensitivity analysis, where we evaluated robustness towards noise, occlusion and clutter. Overall our findings were that all feature performance drop with increasing noise, occlusion and clutter. The most unstable feature towards noise is FPFH and more robust is USC. Compared to local histogram features PPF performance degrades faster under increasing occlusion and clutter levels, but it still stays higher for most of the datasets. Overall datasets, the most frequent top performer is PPF, then SHOT and USC features. The highest mean AUC (area under the PR curve) is achieved by SHOT feature, then follows USC and PPF.

In the end of the work, we presented object detection and pose estimation results using a RANSAC algorithm on top of the found point correspondences. We chose to use RANSAC also for PPF instead of originally proposed voting scheme by Drost et al. (2010) in order to be able to compare PPF performance with local histogram features on a feature level and not to bias our results on the object recognition algorithm. However, in order to see the performance difference between RANSAC pose estimation and voting for the pose, we included object detection and pose estimation results for the original PPF implementation (PPF$_{voting}$). We used the implementation of PPF$_{voting}$ from the commercial product HALCON.

Most of the achieved object detection and pose estimation results were consistent with the PR results - higher precision/recall values resulted in higher detection rates. There were some exceptions, for example for the PPF$_{colour}$ feature fewer objects were recognised correctly compared to PPF, which was not expected from the PR result. PPF$_{voting}$ outperformed PPF, which leads to the conclusion that it is better to use PPFs in a voting based pose estimation rather than in a RANSAC pipeline. Overall PPFs perform best on data with high levels of

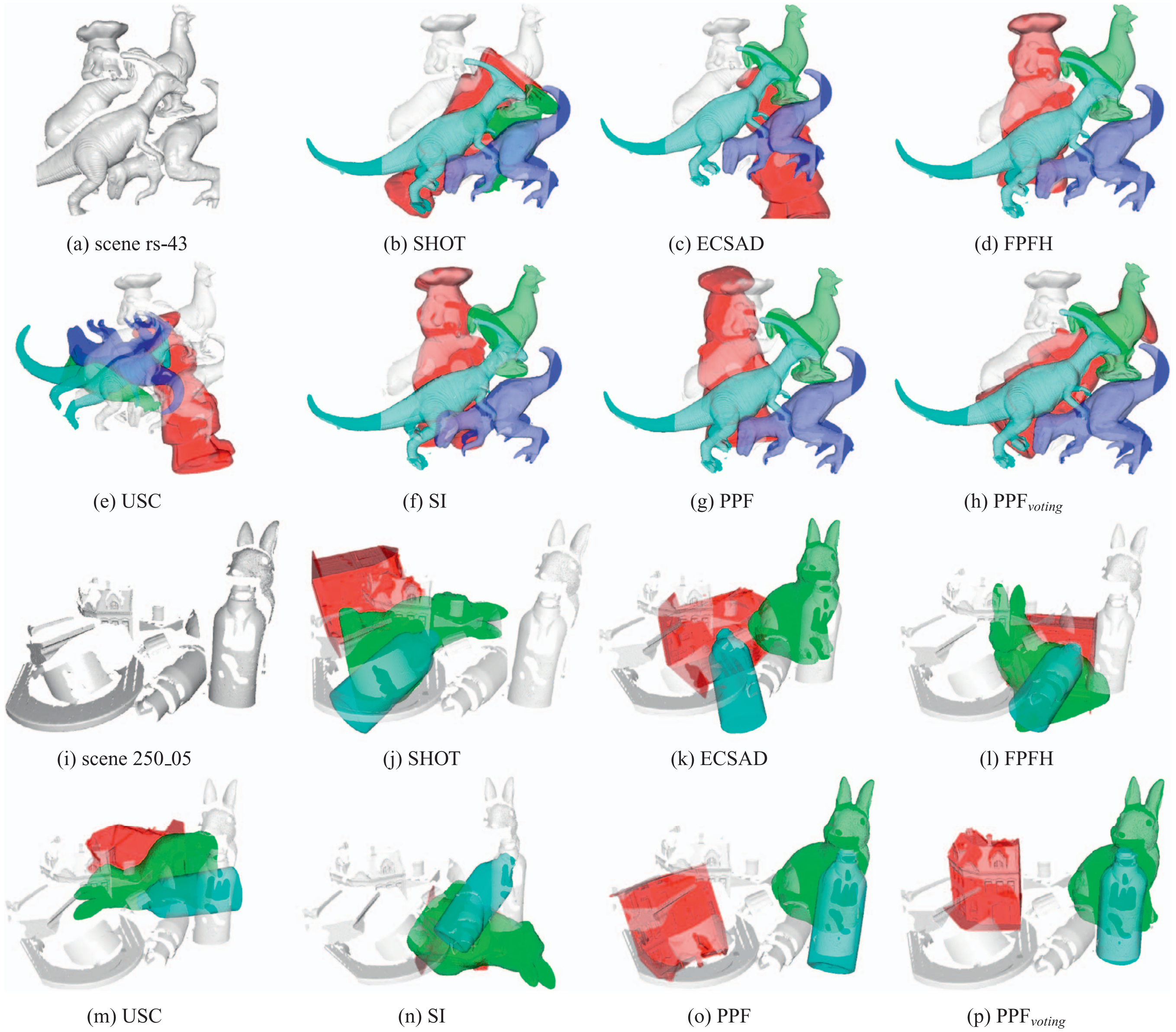


Fig. 17. Object detection results for UWA (scene rs-43 (a–h)) and DTU (scene 250_05 (i–p)) datasets. For the UWA dataset, we get successful recognition for FPFH and SI features for all four objects. For the DTU dataset, all three objects are detected correctly by PPF$_{voting}$ and two objects by PPF.

Table 7
The feature descriptor average computation and matching speed per one reference point. The numbers represent $10^{-6}s$. Please note that not all of the features were optimised for faster performance and the numbers are provided for a reference.

Feature descriptor	Feature computation speed	Feature matching speed
SHOT	197	59
ECSAD	219	44
FPFH	6663	14
USC	24,033	240
SI	372	62
PPF	0.48	9724
PPF colour	0.62	12,540

occlusions, conversely local histogram features provide more robust features in scenes with high degrees of noise, but moderate occlusions.

In this work, we have not looked into feature computation speed in depth. We believe that with the rapid change of hardware and if a proper amount of time is put into implementing features efficiently, most of the features can be made faster. Regarding our tests, the PCL versions of SHOT/ECSAD/SI are among the faster ones, while FPFH and USC are slower (as shown in Table 7). The speed of the PPF version presented in this work is rather slow, but it gets significantly faster with extra cores. The bottleneck is the feature matching, which can be speeded up using approximate searches or hash tables as in Drost et al. (2010). The PPF$_{voting}$ is rather fast and can be speeded up by using fewer reference points. Choi and Christensen (2016) showed that by using GPU, PPF_{colour} can be computed/matched more efficiently and much faster than we showed in this paper.

The overall conclusions of our experiments are:

- the distance between two points is the strongest feature in the PPF descriptor;
- the 4-dimensional PPF descriptor is the best choice for most of the data sources;
- if the colour information between object and scene is calibrated, adding colour to the 4-dimensional PPF will increase the descriptiveness of the feature;
- for most of the datasets, PPFs in comparison to local histogram

features have the highest recall values;

- PPF performance significantly decreases with increasing noise, occlusion and clutter, but on an absolute scale it is a stronger descriptor than the used local histogram features;
- due to lower precision of the top ranked matches and overall high recall values, more robust algorithms need to be used with the PPFs (e.g. voting);
- even in a RANSAC object detection pipeline, PPFs correctly detect the highest number of objects;

Acknowledgements

We would like to acknowledge the providers of all the datasets used in this work: University of Bologna, The University of Western Australia, Queen's University, and Technical University of Denmark. The research leading to these results has received funding from the EU FoF Project ReconCell (project number 680431).

References

Bayramoglu, N., Alatan, A.A., 2010. Shape index sift: Range image recognition using local features. Pattern Recognition (ICPR), 2010 20th International Conference on. IEEE, pp. 352–355.

Besl, P.J., McKay, N.D., 1992. Method for registration of 3-d shapes. Robotics-DL Tentative. International Society for Optics and Photonics, pp. 586–606.

Birdal, T., Ilic, S., 2015. Point pair features based object detection and pose estimation revisited. 2015 International Conference on 3D Vision. IEEE, pp. 527–535.

Buch, A.G., Petersen, H.G., Krüger, N., 2016. Local shape feature fusion for improved matching, pose estimation and 3d object recognition. SpringerPlus 5 (1), 1.

Choi, C., Christensen, H., 2012. 3d pose estimation of daily objects using an rgb-d camera. 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 3342–3349.

Choi, C., Christensen, H.I., 2016. Rgb-d object pose estimation in unstructured environments. Rob. Auton. Syst. 75, 595–613.

Choi, C., Taguchi, Y., Tuzel, O., Liu, M.-Y., Ramalingam, S., 2012. Voting-based pose estimation for robotic assembly using a 3d sensor. IEEE International Conference on Robotics and Automation. pp. 1724–1731.

Choi, C., Trevor, A.J., Christensen, H.I., 2013. Rgb-d edge detection and edge-based registration. 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, pp. 1568–1575.

Davis, J., Nehab, D., Ramamoorthi, R., Rusinkiewicz, S., 2005. Spacetime stereo: a unifying framework for depth from triangulation. IEEE Trans. Pattern Anal. Mach. Intell. 27 (2), 296–302.

Drost, B., Ilic, S., 2012. 3d object detection and localization using multimodal point pair features. 2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission. pp. 9–16.

Drost, B., Ulrich, M., Navab, N., Ilic, S., 2010. Model globally, match locally: Efficient and robust 3d object recognition. 2010 IEEE Conference on Computer Vision and Pattern Recognition. pp. 998–1005.

Frome, A., Huber, D., Kolluri, R., Bülow, T., Malik, J., 2004. Recognizing objects in range data using regional point descriptors. European conference on computer vision. Springer, pp. 224–237.

Guo, Y., Bennamoun, M., Sohel, F., Lu, M., Wan, J., 2014. 3d object recognition in cluttered scenes with local surface features: a survey. IEEE Trans. Pattern Anal. Mach. Intell. 36 (11), 2270–2287.

Guo, Y., Bennamoun, M., Sohel, F., Lu, M., Wan, J., Kwok, N.M., 2016. A comprehensive performance evaluation of 3d local feature descriptors. Int. J. Comput. Vis. 116 (1), 66–89.

Hinterstoisser, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G., Konolige, K., Navab, N., 2012. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. Asian Conference on Computer Vision. Springer, pp. 548–562.

Hinterstoisser, S., Lepetit, V., Rajkumar, N., Konolige, K., 2016. Going further with point pair features. European Conference on Computer Vision. Springer, pp. 834–848.

Johnson, A.E., Hebert, M., 1999. Using spin images for efficient object recognition in cluttered 3d scenes. IEEE Trans. Pattern Anal. Mach. Intell. 21 (5), 433–449.

Jørgensen, T.B., Buch, A.G., Kraft, D., 2015. Geometric edge description and classification in point cloud data with application to 3d object recognition. Proceedings of the 10th International Conference on Computer Vision Theory and Applications.

Kiforenko, L., Buch, A.G., Krüger, N., 2015. Object detection using a combination of multiple 3d feature descriptors. International Conference on Computer Vision Systems. Springer, pp. 343–353.

Kim, E., Medioni, G., 2011. 3d object recognition in range images using visibility context. Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on. IEEE, pp. 3800–3807.

Knopp, J., Prasad, M., Willems, G., Timofte, R., Van Gool, L., 2010. Hough transform and 3d surf for robust three dimensional classification. Comput. vis.–ECCV 2010 589–602.

Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. 60 (2), 91–110.

Mian, A.S., Bennamoun, M., Owens, R., 2006. Three-dimensional model-based object recognition and segmentation in cluttered scenes. IEEE Trans. Pattern Anal. Mach. Intell. 28 (10), 1584–1601.

Rusu, R.B., Blodow, N., Beetz, M., 2009. Fast point feature histograms (fpfh) for 3d registration. IEEE International Conference on Robotics and Automation. pp. 3212–3217.

Siddiqi, K., Shokoufandeh, A., Dickenson, S.J., Zucker, S.W., 1998. Shock graphs and shape matching. Sixth International Conference on Computer Vision. pp. 222–229.

Sølund, T., Buch, A., Krüger, N., Aanaes, H., 2016. A large-scale 3d object recognition dataset.

Steder, B., Rusu, R.B., Konolige, K., Burgard, W., 2010. Narf: 3d range image features for object recognition. Workshop on Defining and Solving Realistic Perception Problems in Personal Robotics at the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS). 44.

Sun, J., Ovsjanikov, M., Guibas, L., 2009. A concise and provably informative multi-scale signature based on heat diffusion. Computer Graphics Forum. 28. Wiley Online Library, pp. 1383–1392.

Taati, B., Bondy, M., Jasiobedzki, P., Greenspan, M., 2007. Variable dimensional local shape descriptors for object recognition in range data. 2007 IEEE 11th International Conference on Computer Vision. pp. 1–8.

Tombari, F., Salti, S., Di Stefano, L., 2010. Unique shape context for 3d data description. Proceedings of the ACM Workshop on 3D Object Retrieval. pp. 57–62.

Tombari, F., Salti, S., Di Stefano, L., 2010. Unique signatures of histograms for local surface description. European Conference on Computer Vision. Springer, pp. 356–369.

Tuzel, O., Liu, M.-Y., Taguchi, Y., Raghunathan, A., 2014. Learning to rank 3d features. European Conference on Computer Vision. Springer, pp. 520–535.

Wahl, E., Hillenbrand, U., Hirzinger, G., 2003. Surflet-pair-relation histograms: A statistical 3d-shape representation for rapid classification. Proceedings of International Conference on 3-D Digital Imaging and Modeling. pp. 474–481.

Wu, H.-Y., Zha, H., Luo, T., Wang, X.-L., Ma, S., 2010. Global and local isometry-invariant descriptor for 3d shape comparison and partial matching. 2010 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 438–445.