IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 3 April 2024, accepted 8 May 2024, date of publication 13 May 2024, date of current version 20 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3400591

RESEARCH ARTICLE

FPFH Revisited: Histogram Resolutions, Improved Features, and Novel Representation

JÁNOS MÁRK SZALAI-GINDL AND DÁNIEL VARGA
Faculty of Informatics, Department of Information Systems, Eötvös Loránd University (ELTE), 1117 Budapest, Hungary
Corresponding author: János Márk Szalai-Gindl (szalaigindl@inf.elte.hu)

The work of János Márk Szalai-Gindl was supported by the New National Excellence Program of the Ministry for Culture and Innovation from the Source of the National Research, Development and Innovation Fund under Grant ÚNKP-22-4.

ABSTRACT 3D local descriptors are essential for many computer vision-related tasks: point cloud registration, object recognition, etc. The first descriptors appeared decades ago, but new methods are still emerging today. To solve the above mentioned problems efficiently, we need robust and distinctive descriptors that are invariant to translation, rotation and normal orientation. In this paper, we revisit and analyze the well-known hand-crafted Fast Point Feature Histogram (FPFH) and the three geometric features underlying the FPFH. Modifications to the geometric features are proposed to increase the descriptiveness and improve robustness against inconsistent normal orientation. Thus, normal orientation methods with high computational complexity can be omitted and the performance of the descriptor is nearly doubled. We also propose a novel approach to representing descriptors: using cumulative distribution functions (CDF) instead of histograms. We mathematically analyze the point-pair features and their CDFs, focusing on a special case where two planes intersect. We use this knowledge to approximate functions on real point cloud data sets in the general case. CDFs can be represented by fitted parameters. By storing these parameters, which can be understood as a form of compression, the storage requirements can be reduced while maintaining the descriptiveness.

INDEX TERMS 3D point cloud, feature descriptor, distribution function, curve fitting, normal orientation.

I. INTRODUCTION

In recent decades, sensors that can collect depth data have become cheaper and more widespread. In the past Microsoft Kinect and Intel RealSense sensors were popular, but nowadays, such sensors are also present in many mobile devices used by ordinary users. The data collected by these sensors are widely represented as 3D point clouds.

With the spread of 3D point clouds, many problems have emerged: object recognition [1], pose estimation [2], [3], pairwise registration [4] object classification [5] etc. To solve these problems, local feature descriptors representing a local geometry around a base point have been proposed. The feature descriptors produce feature vectors, which are usually multidimensional vectors containing real values [4], [6], but there are also binary feature vectors [7], [8]. Feature descriptors are designed so that the distance between feature vectors of similar surfaces is small, while the feature vectors of different surfaces are far apart. Feature vectors are used to generate point correspondences based on nearest neighbor searches, which are necessary to solve the above mentioned tasks.

Over the last three decades, many local feature descriptor methods have appeared. In recent years, learning-based methods [9], [10], [11], [12], [13], [14], [15] have gained popularity alongside traditional hand-crafted methods. One of the most influential and widely known methods is the Fast Point Feature Histogram (FPFH) [4]. The reason behind its popularity is that the FPFH is computationally efficient, has good descriptive performance, and has an easily accessible open-source implementation in the Point Cloud Library (PCL).

Over the years, several review papers were published (e.g. [16], [17]), as also performance evaluations on different

The associate editor coordinating the review of this manuscript and approving it for publication was Kumaradevan Punithakumar.

datasets [18], [19], [20]. However, only a few studies have been published that have analyzed the geometric features not only empirically but also mathematically. For example, Jutzi and Gross [21] used the eigenvalues of a covariance matrix calculated for the local neighborhood of a point to classify points. To do this, they mathematically determined the eigenvalues associated with certain geometric primitives (line, plane, half-plane, etc.).

In our work, we investigate the point-pair features proposed by Wahl et al. [22] and used in FPFH and several subsequent works. On the one hand, we are interested in whether there is a way to get rid of the costly normal orientation step that the original FPFH needs to avoid having an artifact that distinguishes between points on the same or similar surface just because their estimated normal vectors are in opposite directions. On the other hand, we would also like to explore the question of whether there could be a more concise representation than the histogram for FPFH features computed around a point. Our contribution is as follows:

- We propose a modification to the point-pair features that form the basis of the FPFH feature descriptor. By using the new features, the descriptiveness of the original FPFH is nearly doubled.
- We evaluate the FPFH descriptor with different histogram resolution (number of bins) with both the original and modified point-pair features.
- Instead of representing the local environment with a histogram, we recommend using the cumulative distribution function (CDF). With the parameters associated with the CDF representation, (almost) the same information can be stored in less space, which can be seen as a form of compression.
- The use of CDFs is supported by detailed theoretical considerations, which are presented in the appendices. In our work, we only consider the FPFH features, but the methodology can be applied to other features.

The rest of this paper is structured as follows. Section II reviews the related works, focusing on earlier local feature descriptors. Section III-A presents a brief introduction to the FPFH feature descriptor. Section III-B gives a detailed explanation of point-pair feature modifications, while III-C introduces the CDF representation of the features. Section IV explains the methodology of the evaluation, and Section V presents the results of our evaluation. The conclusions and our plans for the future are in Section VI. In Appendix A, we prove that the modified features are invariant to normal orientation. Appendix B presents a detailed explanation of CDFs, including mathematical theorems and proofs related to them. Appendix C shows the deduction of the Hellinger distance to quantify the similarity between two fitted CDFs of FPFH features.

II. RELATED WORK

The first local feature descriptors for 3D objects appeared more than three decades ago. One of the first popular works is that of Stein and Medioni [23] (TOSS), who called the local environment of a point a ''splash'' and used the normal vector of neighbor points to characterize the local environment. The work of Chua and Jarvis [24] computes a feature vector based on the distance between the points in the local environment and the plane defined by the base point. This approach is similar to the later published local feature descriptor methods. In their work, Hetzel et al. [25] used distance, normal vectors and principal curvature to calculate local feature histograms, and performed object recognition by matching histograms, very similar to later methods.

In 1999, the feature descriptor known as Spin Images (SI) [26] was proposed. It creates a histogram by rotating a plane. In the paper, they evaluated the effect of varying the size of the local environment and the importance of the bin size of the histogram. Their process for object recognition is very similar to the feature-based registration processes that later became popular [27]. The importance of SI is indicated by the fact that it is almost always mentioned in review papers summarizing the descriptors [18], [19].

Over the years, a number of well-known feature descriptors appeared. In 2004, Frome et al. [28] introduced a method called 3D Shape Context (3DSC), which was an extension of a two-dimensional method. The space around the base point is partitioned along azimuth, radial, and elevation directions, and the resulting bins contribute to the elements of the descriptor vector. The method was improved by Tombari et al. [6], by introducing a Local Reference Frame (LRF), making the descriptor invariant to rotation. They called the resulting new descriptor Unique Shape Context (USC). In their work, Salti et al. [29] proposed the Signature of Histograms of OrienTations (SHOT) descriptor, which partitions the space around the base point into bins similar to USC. They put a strong emphasis on the definition of the Local Reference Frame (LRF). Unlike USC, the SHOT descriptor creates a histogram based on the normal vectors of the points in each bin. The final feature vector is constructed by appending these histograms. In the same work, they classify local descriptors into two groups: histogram and signature-based. Another well-known descriptor is Rotational Projection Statistics (RoPS) [30], which projects 3D points onto two-dimensional planes. By dividing the planes into bins, matrices are obtained. The feature vectors are constructed from the matrices based on Shannon entropy and moment theory. Most of the above methods are implemented in the Point Cloud Library (PCL) [31].

In 2008, Rusu et al. [32] proposed a new descriptor called Point Feature Histogram (PFH). The descriptor makes point pairs using the points in the neighborhood of the base point. For each point pair, four features are computed, which are used to generate histograms. The concatenated histogram values give the final feature vector. The point-pair features used in this paper are based on previous work by Wahl et al. [22]. These features are calculated using the point coordinates and their normal vectors. Later, Rusu et al. [4] improved their method. They omitted one feature which represents the distance between points and created fewer

point-pairs per base point. Thus the computational complexity became linear instead of quadratic. The new descriptor is called Fast Point Feature Histogram (FPFH). Over the years, many works appeared comparing the performance of local feature descriptors using different data sets for different use cases [18], [19], [20]. In general, FPFH achieved the best results in most cases. That is one of the reasons why this method has become widely known and used. The authors implemented the PFH and FPFH descriptors in PCL.

A few years later, feature binarization methods and binary feature descriptors appeared. A binarization method replaces a real-valued element with one or more binary values in a feature vector [33], [34]. In contrast, binary feature descriptors produce feature vectors that contain binary values by default [7], [8]. New local feature descriptors have also appeared in recent years.

The KDD descriptor [35] differs from the previous ones in that it does not use a spherical neighborhood around the base point but a cubic. In their work, Prakhya et al. [36] propose a new descriptor, 3DHoPD, which consists of two parts. The first three elements of the descriptor represent the coordinates of the base point in a transformed three-dimensional space, and the following 15 elements describe the local geometry. Thus, a two-phase matching can be performed: first, the feature vectors are filtered based on the first three elements, and only the second phase uses the other elements.

Many other local feature descriptors use very similar features as FPFH [1], [37]. Drost et al. [2] proposed a different approach but also used similar features to characterize point pairs. Their method is called Point Pair Feature (PPF), and they construct a global model descriptor from point pairs of an object. The basic idea was further developed in later works, but the point-pair features remained the same throughout [3], [38]. From the above works, it is clear that the features used by the FPFH [4] method describe well the geometric relationships of the points.

The works mentioned above are so-called hand-crafted descriptors. The hand-crafted descriptors published in recent years do not receive as much attention because of the newly developed learning-based local descriptors. One of the first learned local descriptors was 3DMatch [39] in 2016. In the following years, several works were published [9], [10], [11], [12], [13], [14], [15]. Many comprehensive reviews of hand-crafted and learning-based methods has been published since then [16], [17]. Learning-based methods achieve better results than hand-crafted methods, however, a learning process is necessary, and they work well when used on similar data like was used for training.

The relevance of the FPFH method is illustrated by the fact that it is still compared with recent learning-based methods. The point-pair features it uses are also used in many subsequent works [1], [37]. Our aim was to study the point-pair features used by the FPFH method not only empirically but also theoretically.

III. METHOD

In Section II, we tried to highlight the importance of FPFH. This section contains, on the one hand, a presentation of the main components that are relevant to it and, in some respects, a further development of the original approach. On the other hand, we investigate how to determine the CDFs of features and estimate their associated parameters instead of aggregating a pair of features from a point and its neighborhood into a histogram. The storage of the estimated parameters can be seen as a kind of information compression with respect to the original histogram.

A. FPFH WITH ORIGINAL FEATURES

As indicated in the introduction, the FPFH point-pair features [4] presented here are essentially adapted from Wahl et al. [22]. For this reason, both works serve as a basis for the following description, where $\cdot$ denotes the dot product, $\times$ the cross product and $\| \bullet \|$ the Euclidean norm.

Suppose that we want to capture the properties of the local structure of the point cloud in radius R around point p_1 which we call the base point. Let p_2 denote a neighbor point for which $0 < \|p_2 - p_1\| \leq R$. Let the estimated surface normal vectors be calculated for each point, denoted by n_1 and n_2. Define a Darboux frame [40] for the pair of points p_1, p_2 as follows, where point p_1 is the origin if the following is true (otherwise point p_2 is chosen):

$$|n_1 \cdot (p_2 - p_1)| \leq |n_2 \cdot (p_2 - p_1)| \tag{1}$$

and the orthonormal basis will be (provided that p_1 is the origin)

$$u = n_1, \tag{2}$$

$$v = \frac{(p_2 - p_1) \times u}{\|(p_2 - p_1) \times u\|}, \tag{3}$$

$$w = u \times v. \tag{4}$$

The point-pair features will be as follows, keeping the notation of the original FPFH paper [4]:

$$\theta = \arctan(w \cdot n_2, u \cdot n_2),$$
$$\alpha = v \cdot n_2,$$
$$\phi = \frac{u \cdot (p_2 - p_1)}{\|p_2 - p_1\|}.$$

In order to imagine the features easily, they are illustrated in Figure 1, where the translation of the normal vector n_2 to the origin is denoted by n_2', and vector n_2^{uw} is the projection of n_2' onto the uw-plane.

The FPFH descriptor produces a feature vector by the following steps. What we have seen so far are only the FPFH features computed for a given point p_1 and its neighbor point p_2. This computation is performed with all the R-neighborhood points p_1: $\{p_2, \ldots, p_{N+1}\}$ (N is the number of neighbors) and the resulting values θ, α and ϕ are recorded.

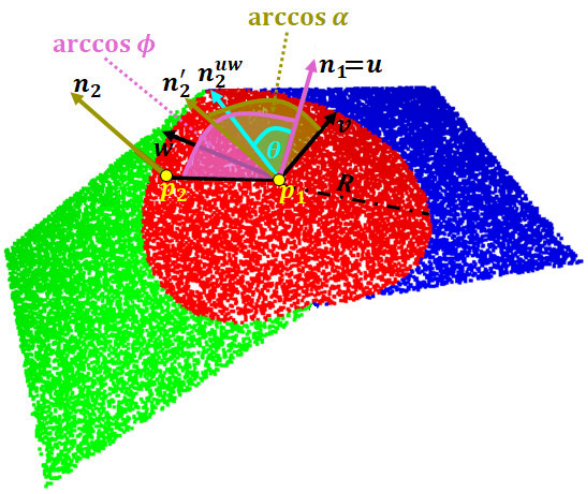


FIGURE 1. An illustration of the FPFH features for the pair of points p_1, p_2. See text for an explanation of the symbols.

Then, per feature, the ranges

$$\begin{aligned} \theta &\in [-\pi, \pi], \\ \alpha &\in [-1, 1], \\ \phi &\in [-1, 1] \end{aligned} \tag{5}$$

are divided into BIN non-overlapping intervals, in other words, bins, and the number of feature values that fall into each bin is counted. The resulting numbers are referred to below as bin weights. We will see later in Section V that the choice of BIN plays an important role in terms of descriptiveness.

We obtain the so-called Simplified Point Feature Histogram of point p_1, denoted by SPFH(p_1), by concatenating the histograms obtained per feature in the previous step into a vector. In order to obtain the final FPFH(p_1) feature vector, the SPFH(p_i) values for the p_i neighbor points ($i = 2, \ldots, N+1$) should also be computed and the bin weights should be divided by the distance of p_i from p_1. The following formula is obtained

$$\text{FPFH}(p_1) = \text{SPFH}(p_1) + \frac{1}{N} \sum_{i=2}^{N+1} \frac{\text{SPFH}(p_i)}{\|p_i - p_1\|}. \tag{6}$$

Note that the FPFH implementation of the PCL library [31] mentioned above and the Open3D library [41], also widely used, includes a normalization part per feature histogram, the output of which is a transformation of the bin weights (divided by distance) so that their sum is 100 and the original ratios are preserved. In the PCL implementation, the term SPFH(p_1) is missing from the above formula. However, it is included in the Open3D implementation and its bin weights are normalized in a similar way. To summarize, FPFH(p_1) will be a vector of dimension $3 \cdot$ BIN, where for θ

$$\sum_{d=0}^{\text{BIN}-1} \text{FPFH}(p_1)[d] = 100 \text{ (PCL) or } 200 \text{ (Open3D)},$$

for α

$$\sum_{d=\text{BIN}}^{2\,\text{BIN}-1} \text{FPFH}(p_1)[d] = 100 \text{ (PCL) or } 200 \text{ (Open3D)},$$

and for ϕ

$$\sum_{d=2\,\text{BIN}}^{3\,\text{BIN}-1} \text{FPFH}(p_1)[d] = 100 \text{ (PCL) or } 200 \text{ (Open3D)}$$

where FPFH(p_1)[d] denotes the d^{th} coordinate of the feature vector.

B. MODIFIED FEATURES

The FPFH features are highly dependent on the accuracy of the normal vector estimation. Several solutions to this problem have been proposed, for a comparison of these methods we refer the reader to [42]. We note that more recently deep learning normal estimation methods are also starting to be used (see e.g. [43], [44]). Besides the accuracy of the normal vectors, it is also important that the methods pay attention to the fact that the normal vectors belonging to points close to each other are preferably in the same direction, i.e., that their orientation is consistent. A popular method for this task is the work of Hoppe et al. [45]. In the past few years many learning-based [46], [47] and other novel methods [48] have appeared to solve this task. It is important to note that achieving globally consistent normal orientation is time-consuming task. In the method of Hoppe et al. [45], the algorithmic complexity of constructing a Minimal Spanning Tree is $\mathcal{O}(n \log n)$. The high computation time is also a problem for new methods. The algorithm proposed by Xu et al. [48] achieves better results than previous works, but the algorithm scales non-linearly and can run for several minutes for point clouds of a few thousand points (on a single CPU machine).

However, even in the case of consistently oriented normal vectors, anomalies can occur in FPFH vectors, which are discussed below. Consider Figures 2 and 3 to illustrate the problem. Figure 2 shows simulated point clouds where two planes are at the same angle to each other. The normal vectors corresponding to the points are shown by black arrows. Except for the point cloud shown in Figure 2c, all point clouds have been oriented. Furthermore, a point p is marked on the figures, for which the vector FPFH(p) is computed (it is located in almost the same position for each point cloud). For each point cloud there is a color code in the upper right corner, which can be used to identify the weights of the associated FPFH(p) bins in Figure 3. It can be seen that for the bins belonging to features θ and ϕ (i.e., at the two edges), the bin weights do not match for either of the two point clouds. It can therefore be seen that, even though normal orientation has

been applied, it does not help that points at the same position have the same feature vector in point clouds containing planes with the same angle. It is important to note that, although it is a good idea to orient the normal vectors towards one viewpoint, there are two cases where this is true but the FPFH vectors do not match (for example, because we are looking at the surface from the other side). This in turn can make it difficult to find true correspondences between two point clouds.

The original FPFH feature definitions should therefore be reviewed to see if they could be improved with modifications. One important question is whether it is really necessary to swap roles between p_1 and p_2 if condition (1) is not satisfied. To the best of our knowledge, we attempt to reconsider this for the first time. Condition (1) is already stated in the paper [22] on which the FPFH method is based. Already in that paper the authors argue for it, but a slightly more extended explanation can be found in Wahl's dissertation [49]. This shows that they wanted to avoid the case where $(p_2 - p_1) \times u$ is a zero vector in equation (3), because then v would be undefined. This is helped by checking condition (1) in that if $|n_1 \cdot (p_2 - p_1)|$ is (almost) equal to 1 (i.e. n_1 is (almost) collinear with $(p_2 - p_1)$ and hence $(p_2 - p_1) \times n_1$ is (almost) a zero vector), but $|n_2 \cdot (p_2 - p_1)|$ is smaller than this, then it is better to choose p_2 as the origin and n_2 as u.

It is certainly true that the extreme case above could indeed cause problems in principle. However, there is a side effect of this: there will be a swap of roles even when $|n_1 \cdot (p_2 - p_1)|$ is relatively far from 1 and only slightly larger than $|n_2 \cdot (p_2 - p_1)|$. Our tests have shown that this swap impairs descriptive ability (as we will come back to in Section V). If we omit the swap, the feature vectors become more similar, see Figure 4. In this figure, the improvement can already be seen in the bins 22-32 for feature ϕ. For the point clouds marked with blue and red, and orange and purple color codes, respectively, the bin weights are now visibly close to each other. In the light of the foregoing, it would be better to first check whether $|n_1 \cdot (p_2 - p_1)|$ is close to 1 and whether $|n_2 \cdot (p_2 - p_1)|$ is significantly smaller than it (by choosing some thresholds). The detailed elaboration of this is out of scope for the present work, but can be addressed as future work. It is important to note that the possible swap of roles between points p_1 and p_2 based on checking condition (1) will be omitted in the remainder of this paper.

Now the situation looks a little bit better. However, on the one hand, the feature vectors computed for the (blue, red) point cloud pair and the (orange, purple) point cloud pair still do not match on the bins 22-32 (although it can be seen the point clouds can be rotated into each other). On the other hand, no significant improvement has been made for bins 0-10 of the feature θ. However, reflection symmetries can be detected in Figure 4. In the case of the part of the figure relating to the feature θ, we see that if reflection was to occur where the axis of reflection was bin 5, the red and purple and the orange and blue weights, respectively, would roughly overlap. The situation is similar for feature ϕ, where the axis of reflection is obviously bin 27.

A more careful consideration of the above, it can be deduced, see Appendix A, that the orientation of the normal vectors will no longer be of interest after the following extra steps have been taken. If $\phi \leq 0$, then $\phi' = \phi$ and $\theta' = \theta$ (no changes), otherwise

$$\phi' = -\phi, \tag{7}$$
$$\theta' = \arctan(-w \cdot n_2, u \cdot n_2) \tag{8}$$

and after both cases above,

$$\theta' = \begin{cases} \theta' + \pi & \text{if } \theta' < -\pi/2, \\ \theta' - \pi & \text{if } \theta' > \pi/2 \end{cases} \tag{9}$$

and looking at α, if $u \cdot n_2 \geq 0$, then $\alpha' = \alpha$ (no change), otherwise

$$\alpha' = -\alpha, \tag{10}$$

where θ', α' and ϕ' denote the modified features. In the following we will work with modified features, so it is not necessary to distinguish between them and the original features, so we will use the notation θ, α and ϕ for the modified features. Note that after the modification, the ranges per feature will be as follows

$$\begin{aligned} \theta &\in [-\pi/2, \pi/2], \\ \alpha &\in [-1, 1], \\ \phi &\in [-1, 0], \end{aligned} \tag{11}$$

cf. ranges (5).

If we create the FPFH feature vectors with this modification, Figure 5 shows that they are practically identical. (Minimal differences are due to the fact that the feature vectors were computed for different simulated point clouds, where there are small random variations between the point clouds.) Looking at the figure, it can also be seen that even if the normal vectors are not oriented at all, the feature vector is still the same as the others, see the green colored bins. There is therefore no need for the costly orientation computation.

It is important to note that a similar problem arises for other local feature descriptors that need a unique and unambiguous local reference frame (LRF) [6], [29]. In general, these methods use the eigenvectors of the neighborhood covariance matrix around the base point obtained by eigenvalue decomposition (EVD) to give a unique LRF. As we can see in the case of the method proposed by Frome et al. [28], this is not enough to obtain a unique and repeatable LRF. To make the LRF repeatable the sign of the eigenvectors must be oriented. Tombari et al. [6] in their work orient the axes of the LRF in the direction where the majority of the points in the neighborhood are. To do this, they consider the dot product between the vector from the base point to the neighbor point and the axis (eigenvector).

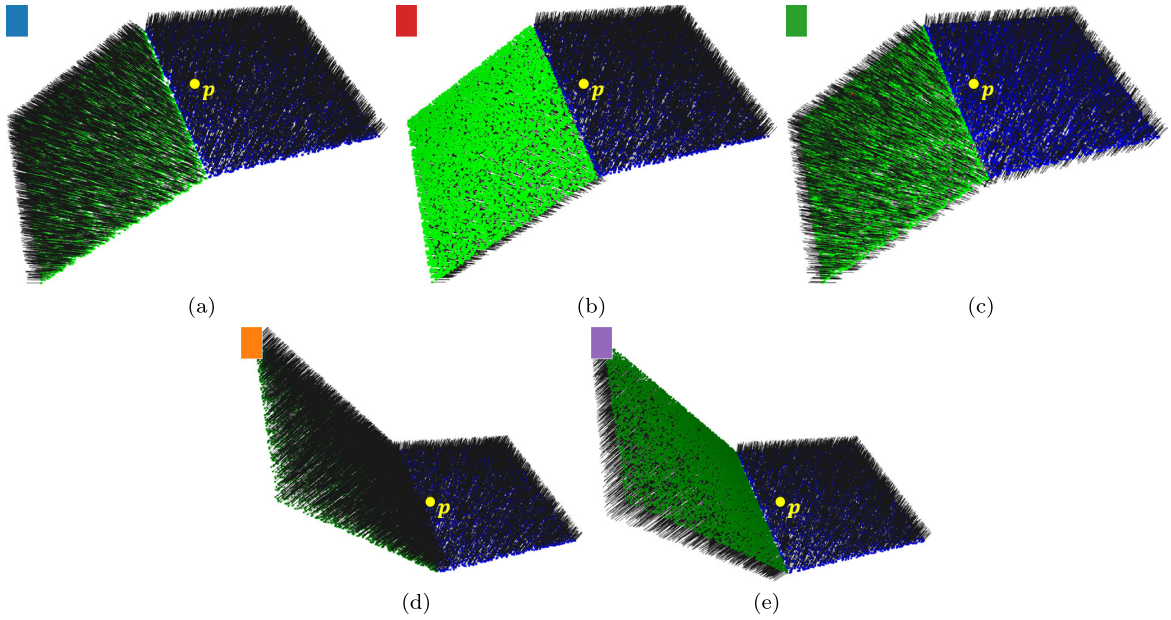


FIGURE 2. Different orientations of the normal vectors of two intersecting planes. The color codes in the upper left corner of the sub-figures help to interpret Figures 3, 4 and 5.

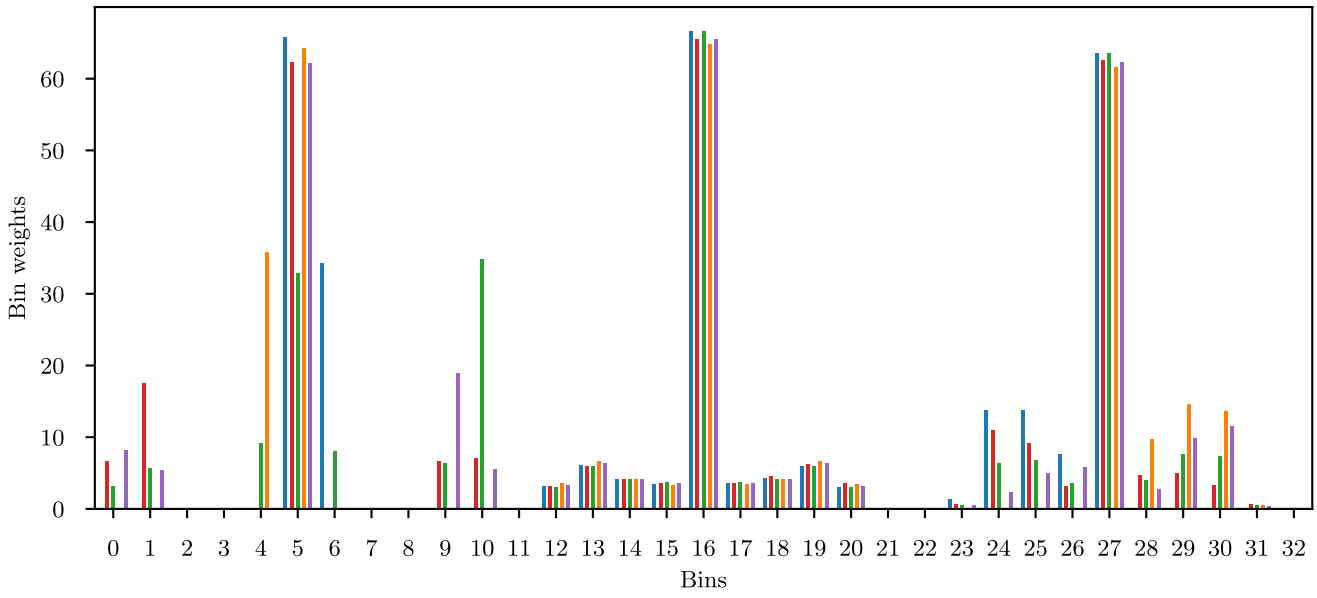


FIGURE 3. The values of the original FPFH feature vectors for *p* points of the different color-coded point clouds in Figure 2. (The colors of the bars follow the colors assigned to the point clouds.).

C. CDF REPRESENTATIONS OF FEATURES

Feature descriptors can be based on different foundations, but the most typical are histogram-based solutions, as underlined by Guo et al. [19] in their paper. FPFH is also a histogram-based descriptor. In this method, we saw that histograms approximate the distribution of features θ, α, ϕ. How accurate this can be also depends on how large the BIN is chosen to be, i.e. what the histogram resolutions will be. (As indicated earlier, we will come back to this in Section V.)

In this session we will propose a new representation that tries to capture these distributions in a different form. We want to attempt to examine the CDFs of the underlying FPFH features and store the parameters of these functions instead of the histogram bin weights as FPFH feature vectors. To the best of our knowledge, this is the first attempt to represent features in this way. We believe that several ideas behind our approach could be applied to other feature descriptors.

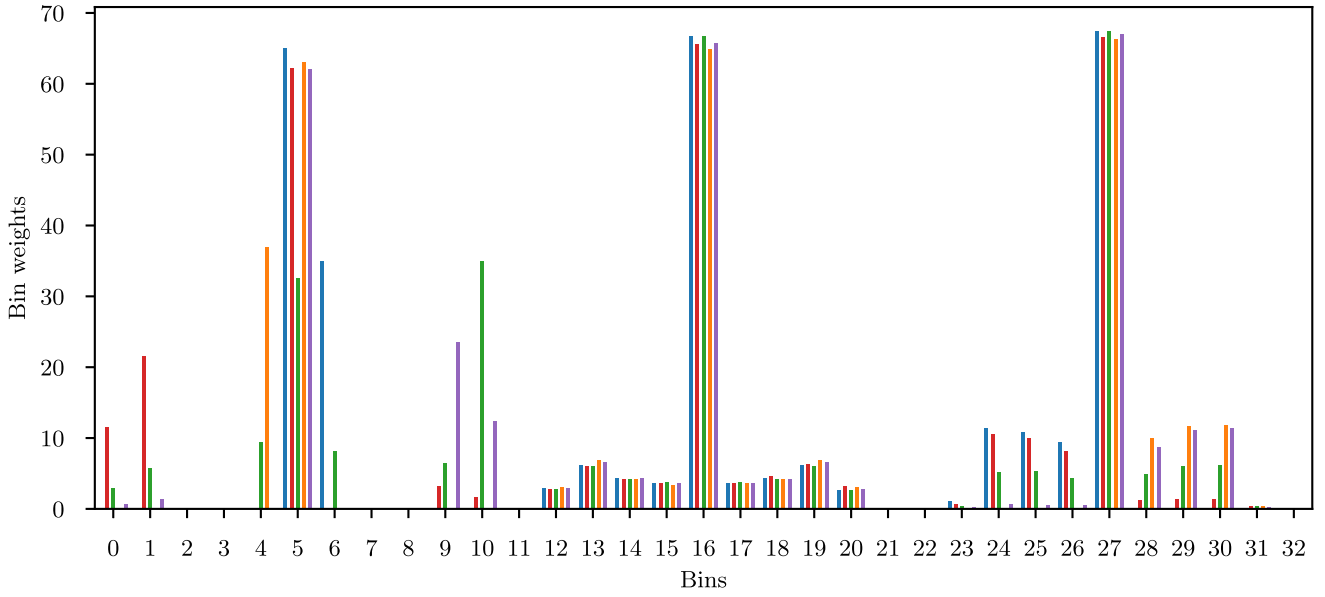


FIGURE 4. The values of the no-role-swapped FPFH feature vectors for p points of the different color-coded point clouds in Figure 2. (The colors of the bars follow the colors assigned to the point clouds.)

In this way, we can in principle provide the same information by storing fewer values. This is therefore somewhat related to the issue of information compression in feature vectors, for which several approaches have been developed: dimensional reduction methods (e.g. [50], [51], [52], [53]), binarization (see Section II) and binary test selection (see e.g. [54] and the references given there). The results of methods using dimensional reduction techniques suggest that one can capture nearly the same information by storing fewer values. This has also motivated the present work.

The other important motivation was that these methods typically work on the basis of different experiments, empirical inferences and have been tested on more or less data sets. Usually, there is no indication of the theory behind them. On the other hand, the more precisely we try to model the world, the more factors we try to include in the model, the more complex and time-consuming the theoretical considerations and derivations become. In our work, we have tried to make our conclusions as independent as possible from the specific data sets used in a particular use case, but the theoretically supported results are still only valid for specific cases. Nevertheless, we could use them in some form for the general case by fitting some appropriate functions, where the "proof" was only empirical. We therefore also say that we have a mathematical conjecture for the general case. The details follow.

In the following, we will examine the FPFH features according to the modifications described in Section III-B. Denote the CDFs of features θ, α, and ϕ with F_θ, F_α, and F_ϕ, respectively.

CDFs are easiest to define when all points in a neighborhood lie in a perfect plane. Then vectors u and n_2 will be parallel (or anti-parallel), so on the one hand $u \cdot n_2$ equals 1 (or -1), and on the other hand $v \cdot n_2$ and $w \cdot n_2$ equal zero (by equations (3) and (4), since vectors v and w are perpendicular to vector u). So $\theta = \alpha = 0$ (if $u \cdot n_2 = -1$ was the case, it is because we used the modification given in equation (9)). Also, since vector $p_2 - p_1$ lies in the plane, vectors u and $p_2 - p_1$ are perpendicular to each other, so $\phi = 0$. In this case, the probability distributions of the features are therefore discrete and it is true that

$$F_\theta(t) = F_\alpha(t) = F_\phi(t) = \begin{cases} 0 & \text{for } t < 0 \\ 1 & \text{for } t \geq 0. \end{cases} \tag{12}$$

For any other type of neighborhood of a base point, it is much more complicated to define CDFs. If the radius is not chosen to be very large, then the neighborhood of a base point can be realistically approximated as the intersection of two or more planes with some accuracy. Suppose that we can somehow define CDFs for the intersection of two planes. Since the features are always computed per pair of points and we can group the neighbor points according to which plane they belong to, the mixture distribution from the CDFs per plane gives the final definition of the functions F_θ, F_α and F_ϕ (the mixture weights are given by the proportions of points in the different planes). Thus, it can be traced back to the case of the intersection of two planes.

Two intersecting planes also has several subcases based on the dihedral angles of the planes. We have examined only one of these types, which occurs frequently in reality, where the dihedral angle is $\pi/2 - \gamma$, $0 \leq \gamma < \pi/2$, in more detail, see Appendix B. In addition, we did not model the effect of noise (we assumed perfect planes). Nevertheless, as we will see, we can still use the results obtained there in the general case. We repeat the main result of Appendix B (equation (60)): with the right parameters, the following functions approximate the

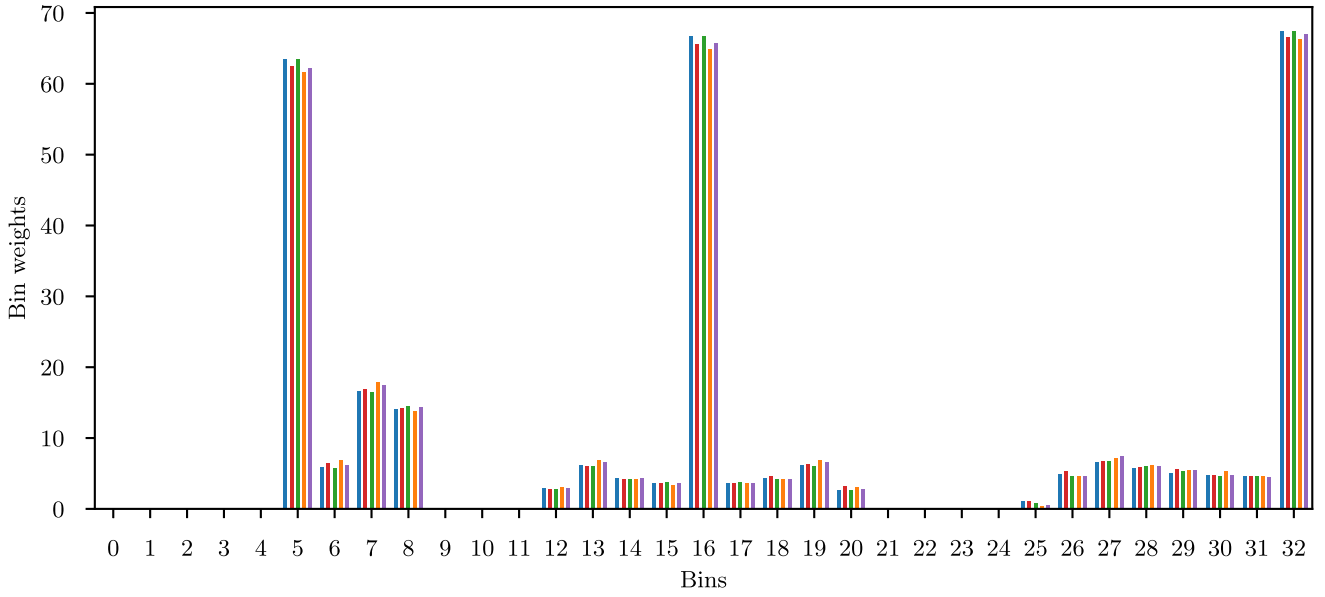


FIGURE 5. The values of the modified FPFH feature vectors for p points of the different color-coded point clouds in Figure 2. (The colors of the bars follow the colors assigned to the point clouds.)

CDFs of the features very well:

$$
\begin{aligned}
B_\theta(t) &= I_{\frac{\arcsin(\tan(t)\tan(\gamma))-\arcsin(\ell/R)}{\pi/2-\arcsin(\ell/R)}}(a,b)\\
B_\alpha(t) &= I_{\frac{\arccos(-t/\cos(\gamma))-\arccos\left(\sqrt{1-\ell^2/R^2}\right)}{\arccos\left(-\sqrt{1-\ell^2/R^2}\right)-\arccos\left(\sqrt{1-\ell^2/R^2}\right)}}(a,b)\\
B_\phi(t) &= 1 - I_{\frac{\arcsin(t/\phi_0)}{\pi/2}}(a,b)
\end{aligned}
\tag{13}
$$

where $I_\bullet(a,b)$ denotes the regularized incomplete beta function with parameters a, b, R denotes the radius of the neighborhood, the base point is located at a distance ℓ from the intersection of the planes and ϕ_0 denotes the minimum value of ϕ (see equation (30)). This was verified by fitting the functions $B_\theta, B_\alpha, B_\phi$ to the empirical CDFs $\widehat{F_\theta}, \widehat{F_\alpha}, \widehat{F_\phi}$, which were obtained from the values θ, α, ϕ of synthetic data sets. (Note that the FPFH is computed using the SPFH values of the neighbor points, see equation (6), something similar to which we did not take into account in Appendix B. Regardless, the resulting formulas work more generally, as will be shown in Section V.)

In the general case, it would be convenient to detect planes in the point cloud and use the formulas for the planes in the vicinity of a base point separately. This would be a time consuming task, but fortunately there is a simpler solution. The idea is that the mixture components can be expected to be somewhat detectable in the mixture CDF curve. The points of the curve where there is a spectacular change in the slope of the function would be good candidates for this, since the values of the CDFs associated with each component will be between 0 and 1 over intervals of different lengths. (As can be seen from Corollary 1 in Appendix B, the actual ranges of feature values depend on the dihedral angle of the two intersecting planes and the distance of the base point from the intersection of the two planes, which are expected to be different from component to component.) How could spectacular changes in the slope of the function be detected? Obviously there is no change in slope at all between two points where the mixture CDF curve is a straight line. Using this fact, we would like to find a point on the curve over an interval that is farthest from the straight line connecting the curve points above the two endpoints of the interval. We call this a knee point if it is farther away from a predetermined distance threshold, otherwise we consider it as if the curve is approximately a straight line for the given interval. Such knee points can be detected recursively on the full mixture CDF curve. First, we take the interval where the mixture CDF takes values between 0 and 1 and determine the knee point, if it exists. Then, we divide our initial interval into two parts along the detected knee point and recursively proceed to detect additional knee points on the resulting two intervals. Note that points where some interesting change occurs in the curve of the function could also be detected by examining the derivatives of the mixture CDFs. (For example, inflection points can be interesting, where the function changes from concave to convex, or conversely.)

A synthetic point cloud is used to illustrate the above, see Figure 6. (This is a very simple example, we have not dealt with, for example, noise or occlusion.) As you can see, the blue plane of the base point p is intersected by two other planes. The dihedral angle of the blue plane and the green (cyan) plane is $\pi/2 - \gamma_1$ ($\pi/2 - \gamma_2$). We take the R-radius neighborhood of base point p, marked in red in the figure. The base point p is located at distance ℓ from the intersection of the planes. Figure 7 shows the empirical CDF $\widehat{F_\alpha}$ where the knee points obtained by the previous recursive method are marked in red (the distance threshold was 0.02). The first

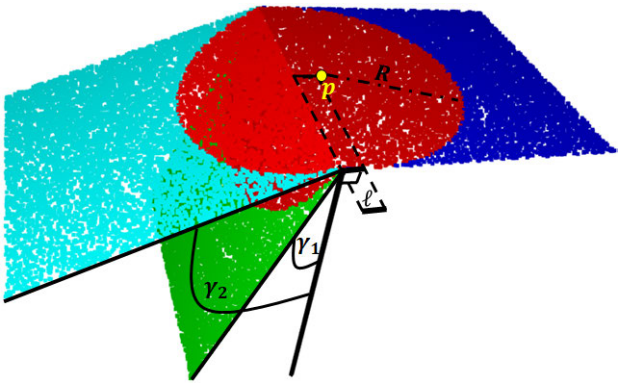


FIGURE 6. The case where the plane of base point p meets two other planes. See text for an explanation of the symbols.

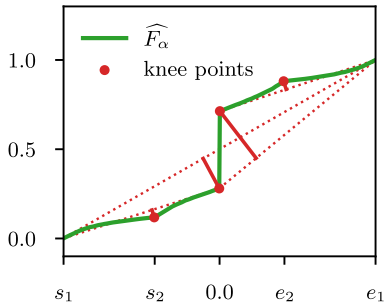


FIGURE 7. The empirical CDF $\widehat{F_\alpha}$ with knee points where the plane of base point meets two other planes. The dotted red lines represent the recursively defined lines from which the farthest points, i.e. the knee points, were sought. s_1 corresponds to $-\sqrt{1-\ell^2/R^2}\cos(\gamma_1)$, e_1 to $\sqrt{1-\ell^2/R^2}\cos(\gamma_2)$, s_2 to $-\sqrt{1-\ell^2/R^2}\cos(\gamma_2)$ and e_2 to $\sqrt{1-\ell^2/R^2}\cos(\gamma_2)$, cf. Corollary 1 in Appendix B.

and the last knee points coincide with the two endpoints of the range of α for the cyan plane (s_2 and e_2 in the figure). This particular synthetic case thus provides some support for the validity of the previous line of reasoning.

Once the knee points (or other interesting points) have been determined, the empirical CDFs $\widehat{F_\theta}, \widehat{F_\alpha}, \widehat{F_\phi}$ must be divided into smaller parts along these points and fitted separately with some modification of the functions given in equation (13). One of the main questions is what to choose instead of the parameters γ, ℓ, ϕ_0 without making complicated investigations on the location of the base point with respect to the planes. What we should definitely pay attention to is that the domain of the arcsin and arccos functions is $[-1, 1]$. So this can be exploited to make the fit meaningful at all. Let t_s, t_e denote the start and end of the interval to which we want to fit the functions.

Let us first examine the function $\widehat{F_\theta}$. Here we have to choose γ such that $|\tan(t)\tan(\gamma)| \leq 1$ is true. Suppose we want to choose γ as positive. Since the function $\widehat{F_\theta}$ is a monotone increasing function, if we choose γ such that $\max(|\tan(t_s)\tan(\gamma)|, |\tan(t_e)\tan(\gamma)|) \leq 1$ holds, then for all t $|\tan(t)\tan(\gamma)| \leq 1$ will hold. If $t_s < t_e < 0$, then $\max(|\tan(t_s)\tan(\gamma)|, |\tan(t_e)\tan(\gamma)|) = |\tan(t_s)\tan(\gamma)|$ and $\tan(t_s)\tan(\gamma) \geq -1$ are required. For simplicity, let us choose γ for which $\tan(t_s)\tan(\gamma) = -1$, i.e.

$$\begin{aligned}\tan(t_s)\tan(\gamma) &= -1\\ \tan(\gamma) &= -\frac{1}{\tan(t_s)}\\ \gamma &= \arctan(-\cot(t_s))\\ \gamma &= \arctan(\tan(t_s+\pi/2))\\ \gamma &= t_s+\pi/2.\end{aligned}$$

If $0 < t_s < t_e$, then, as before, the following choice will be good

$$\begin{aligned}\tan(t_e)\tan(\gamma) &= 1\\ \tan(\gamma) &= \frac{1}{\tan(t_e)}\\ \gamma &= \arctan(\cot(t_e))\\ \gamma &= \arctan(\tan(3\pi/2-t_e))\\ \gamma &= \arctan(\tan(\pi-(t_e-\pi/2)))\\ \gamma &= \arctan(-\tan(t_e-\pi/2))\\ \gamma &= -\arctan(\tan(t_e-\pi/2))\\ \gamma &= \pi/2-t_e.\end{aligned}$$

If $t_s < 0 < t_e$, choose the former or the latter depending on whether $t_e < |t_s|$ or vice versa. If $\gamma = 0$ (this occurs for the first or the last interval, cf. ranges (11)), then instead of t_s or t_e, we compute γ with a t for which $t = t_s + \varepsilon$ or $t = t_e - \varepsilon$ where $\varepsilon > 0$ is a very small number. After defining γ, in order to use the function I properly for the fit, its argument must be transformed such that the values used for the fit fall within $[0, 1]$ (because this is the domain of the function I). So, with the choice of γ above, fit the following function

$$B_\theta^{[t_s,t_e]}(t) = I_{\frac{\arcsin(\tan(t)\tan(\gamma))-\arcsin(\tan(t_s)\tan(\gamma))}{\arcsin(\tan(t_e)\tan(\gamma))-\arcsin(\tan(t_s)\tan(\gamma))}}(a,b) \tag{14}$$

to the part of the function $\widehat{F_\theta}$ on the interval $[t_s, t_e]$. Specifically, before fitting, the value of the function $\widehat{F_\theta}$ at t_s is subtracted and then the resulting function is divided by the value of the function $\widehat{F_\theta}$ at t_e, denoted by h. (This transformation is necessary so that the dependent data to which the fitting is done goes from zero to one, because it makes fitting easier.) If the function $\widehat{F_\theta}$ were possibly constant over the interval, then no fit would be made, but $(a, b) = (0, 0)$ would be used to indicate this (since it would not normally result). In addition to the parameters (a, b), we also store the value of h, so that we can reproduce the approximation of the function $\widehat{F_\theta}$ over the interval $[t_s, t_e]$ by storing these. More precisely, the value h is not stored at the last interval, because the approximation of the function $\widehat{F_\theta}$ must always arrive at 1.

For the function $\widehat{F_\alpha}$, we need to proceed in a very similar way to the previous one. Here we have to choose γ such that it is true that $|-t/\cos(\gamma)| \leq 1$. Assume again that we want

to choose γ to be positive. If $t_s < t_e < 0$, then, for simplicity, let us choose γ for which $-t_s/\cos(\gamma) = -1$, i.e.

$$\begin{aligned} -t_s/\cos(\gamma) &= -1 \\ \cos(\gamma) &= t_s \\ \gamma &= \arccos(t_s). \end{aligned}$$

($-t_s/\cos(\gamma) = 1$ would result in $\gamma = \pi - \arccos(t_s) < \arccos(t_s)$, since $t_s < 0$.) If $0 < t_s < t_e$, then the following choice is a good one, as before

$$\begin{aligned} -t_e/\cos(\gamma) &= 1 \\ \cos(\gamma) &= -t_e \\ \gamma &= \arccos(-t_e) \\ \gamma &= \pi - \arccos(t_e). \end{aligned}$$

If $t_s < 0 < t_e$, choose the former or the latter depending on whether $t_e < |t_s|$ or vice versa. After defining γ, similarly to the previous case, i.e. fit the following function

$$B_\alpha^{[t_s,t_e]}(t) = I_{\frac{\arccos(-t/\cos(\gamma)) - \arccos(-t_s/\cos(\gamma))}{\arccos(-t_e/\cos(\gamma)) - \arccos(-t_s/\cos(\gamma))}}(a,b) \tag{15}$$

to the part of the function $\widehat{F_\alpha}$ on the interval $[t_s, t_e]$ and the corresponding continuation is very similar to the previous case.

Finally, consider the function $\widehat{F_\phi}$. Here, the parameter ϕ_0 must be chosen in a similar way as γ in the previous ones, i.e. $|t/\phi_0| \leq 1$, ϕ_0 must be negative, cf. ranges (11). For the present feature, $t_s < t_e \leq 0$ always holds. Because of the foregoing, $\phi_0 = t_s$ will be a good choice, since it will satisfy $t/\phi_0 \leq 1$ for all t. Fit the following function

$$B_\phi^{[t_s,t_e]}(t) = 1 - I_{\frac{\arcsin(t/\phi_0) - \arcsin(t_e/\phi_0)}{\arcsin(t_s/\phi_0) - \arcsin(t_e/\phi_0)}}(a,b) \tag{16}$$

to the part of the function $\widehat{F_\phi}$ on the interval $[t_s, t_e]$ and the corresponding continuation is very similar to the previous cases.

So, for each feature, we get three values per interval after the fitting: the parameters a and b of the beta distribution and the "height" h of the given part, which does not need to be stored for the last interval, as mentioned above. Figure 8 illustrates that, as the knee points are increased, the fitted function B_α at different intervals becomes increasingly close to the empirical CDF $\widehat{F_\alpha}$. The previous illustration is presented for base point p of a real point cloud (of the 7-Scenes RGB-D redkitchen data set from Section IV-A), see Figure 9.

IV. EVALUATION

A. DATASETS

To evaluate our contribution, we used the 7-Scenes RGB-D redkitchen, [55] and Redwood livingroom [56] datasets. The data sets consist of 60 and 57 clouds. During preprocessing, voxelgrid sampling was used with a voxel size of 0.01 to ensure a consistent density of clouds. After sampling, the average size of a cloud is 100 000 points, but there are smaller and larger clouds. From the point clouds, we select the 45 most overlapping pairs for both data sets. In this case, the overlap of two clouds is always greater than 60% of the total size of the clouds. The radius of FPFH is set to 0.06. This is six times the voxel size, and its ratio compared to the size of the objects is similar to that used in other works. The same radius was used for all measurements. We use the FPFH implementation of the PCL library.

B. DESCRIPTIVENESS

When comparing feature descriptors, we consider how many correct correspondences a method can find between the points of point cloud pairs. To evaluate the performance of a descriptor, we measure descriptiveness, following Guo et al. [19]. It shows the performance of methods with a precision-recall curve (PRC).

To obtain the PRC, we calculate the *precision* and *recall* values for each point cloud pair and aggregate them. For a point cloud pair, the process is as follows. We select points on both point clouds and look for correspondences between them. For each cloud, we select 10% of the points. The feature vectors are calculated for the selected points. After that, we create correspondences using nearest neighbor searches based on the feature vectors of the points in the two point clouds. To filter the correspondences, we use the nearest-neighbor distance ratio (NNDR) based on the work of Lowe et al. [57]. Based on the NNDR method, a correspondence is accepted if the ratio of the distances to the nearest and second nearest neighbors of a point does not exceed a τ threshold.

Let FPFH(p) be the feature vector of a point p from the left cloud of the cloud pair, and FPFH(p_1) and FPFH(p_2) be first and second nearest neighbors of FPFH(p), which are feature vectors of points p_1 and p_2 from the right cloud. A correspondence is kept if

$$\frac{\|\,\text{FPFH}(p) - \text{FPFH}(p_1)\|}{\|\,\text{FPFH}(p) - \text{FPFH}(p_2)\|} > \tau.$$

The *precision* is the ratio of correct correspondences to the total number of correspondences generated. A correspondence is considered correct if, using the ground truth transformation, the distance between two points of the correspondence is less than a specified limit (in the evaluation, we used 0.04, which is less than the radius of the feature descriptor and four times the voxel size).

$$\textit{precision} = \frac{\text{No. correct correspondences}}{\text{No. all correspondences}}$$

The *recall* represents the ratio of correct correspondences to all possible correct correspondences.

$$\textit{recall} = \frac{\text{No. correct correspondences}}{\text{No. all possible correct correspondences}}$$

By varying the value of τ (between 0.7 and 1.0, with 0.1 steps), we obtain different *precision* and *recall* values,

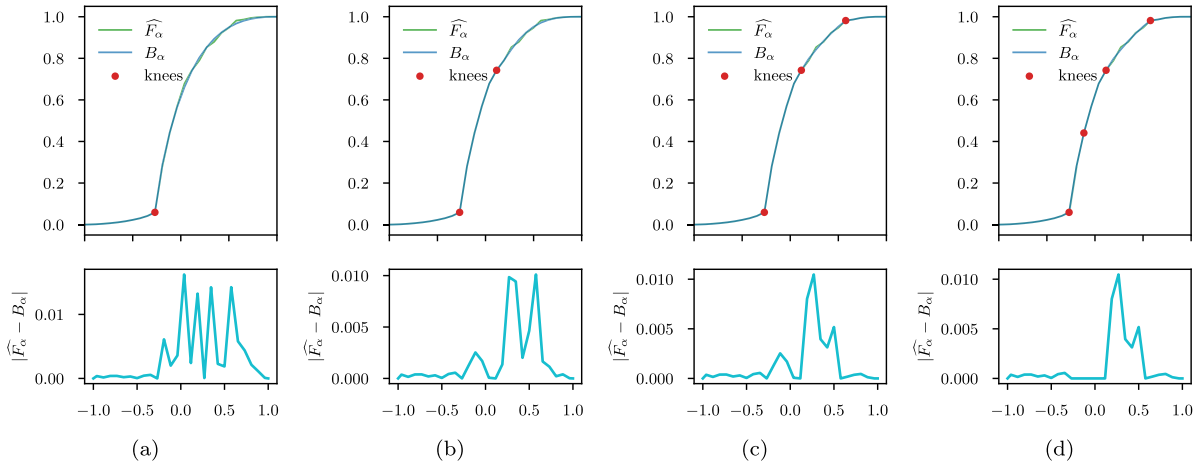


FIGURE 8. Differences between the function $\widehat{F_\alpha}$ and the fitted function B_α for different numbers of knee points. The base point p and its neighborhood, for which the empirical CDF was computed, are shown in Figure 9.

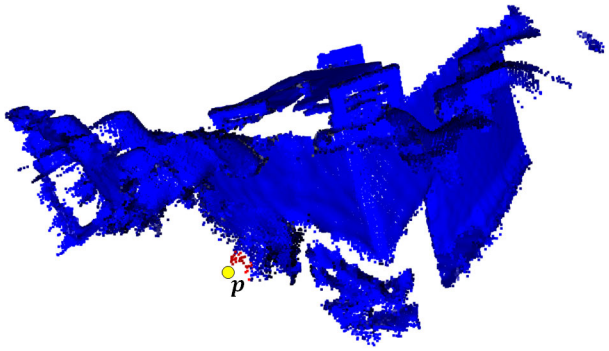


FIGURE 9. A real point cloud of the 7-Scenes RGB-D redkitchen data set (see Section IV-A). The base point p and its neighborhood are marked for which empirical CDF $\widehat{F_\alpha}$ and the fitted function B_α were computed, and these are shown in Figure 9.

which are averaged to obtain the final PCR. The area under the curve (AUC) value is used as a concise representation of the PCR plot, thus providing only one number to denote the descriptiveness of a feature descriptor.

C. FEATURE MATCHING RECALL

The Feature Matching Recall (FMR) is a widely used evaluation metric for 3D point cloud descriptors [9], [11], [13], [15], [58]. The main idea behind this evaluation is that the recall is much more important than the precision. The reason for this is that the precision can be improved by improved correspondence rejection methods. Let $Q = \{q_1, q_2, \ldots, q_n\}$ and $P = \{p_1, p_2, \ldots, p_n\}$ be two 3D point clouds and Ω be the set of correspondences between the points, generated by nearest neighbor (NN) searches in the features space. The average FMR can be calculated for a dataset of matching 3D point cloud pairs as follows:

$$\frac{1}{G}\sum_{g=1}^{G}\mathbb{1}\left(\left[\frac{1}{|\Omega|}\sum_{(p_i,q_j)\in\Omega}\mathbb{1}\left(||p_i - q_jT|| < \tau_1\right)\right] > \tau_2\right)$$

where G is the number of overlapping fragments in the dataset, at least with 30% of overlapping area. $(p_i, q_j) \in \Omega$ is a point-correspondence, $p_i \in P$, $p_j \in Q$, T is the ground truth transformation that aligns Q to P, $||\cdot||$ is the Euclidean distance and $\mathbb{1}$ is the indicator function. The two important parameters of the metric are the τ_1 and τ_2, which denote the inlier distance threshold and inlier ratio threshold respectively. During the evaluation we set $\tau_1 = 0.1$ and $\tau_2 = 0.05$. That means we accept a correspondence as correct if the Euclidean distance between the two points of the correspondence is less than 10cm after aligning using the ground truth transformation and we consider a pair of point clouds alignable if 5% of our correspondences are correct. As with the previous methods, we used a 3DMatch [39] dataset (redkitchen) with 5000 key points for every point cloud, selected by the authors.

D. MANAGING THE FITTED PARAMETERS

The method described in Section IV-B can work with arbitrary metric. The Euclidean distance is used for the traditional FPFH feature vectors. However, in order to determine the descriptiveness of the CDF representation of the features (see Section III-C), it would be nice to use a more sophisticated metric that can work directly with the parameters a, b and h recorded per interval. The Hellinger distance is used to determine the similarity between the CDFs of FPFH features belonging to two points, and the result of Appendix C is that this metric can fulfil the goal of working directly with the fitted parameters. From here on, we denote by PARAMS the descriptor that stores information about the fitted parameters in some form using the Hellinger distance to measure similarity.

It is important to note that the empirical CDFs $\widehat{F_\theta}, \widehat{F_\alpha}, \widehat{F_\phi}$ are in fact a cumulative histogram, where the bin number was 27 (since, as we will see in Section V, for the 7-Scenes RGB-D redkitchen dataset, this bin number was used to achieve the strongest descriptiveness). In addition, the distributions of the point-pair features computed for the neighbor points are taken into account here in a similar way

as the SPFH values of the neighbor points were taken into account in the calculation of the FPFH. These are the basis for the fits of the functions $B_\theta, B_\alpha, B_\phi$ (so, for example, the knee points or other interesting points always fall on the endpoints of the bins). The BOBYQA Algorithm [59] from NLopt library [60] is applied to solve the associated least square problem in the case where the parameter space is bounded (for function fitting, the bounds $0.001 \leq a, b \leq 50.0$ were used in all cases).

It is also important to emphasise that the accuracy of fits depends not only on the bin number chosen, but also on the number of knee points or other interesting points chosen. By adjusting the distance threshold, we can get more or less knee points on the same curve. As we have seen, the more points we take for a particular basis point, the more accurately we can estimate, but the larger the space requirement of the resulting CDF representation. In addition, if knee points are in different locations for each base point, those locations must also be stored. Instead, we examined where the most typical interesting points are in one to two point clouds and selected one such point per feature. For features θ and α, a notable point, 0, was chosen based on several criteria. For ϕ, the choice was made by experimentation and roughly $-1/4$ seemed to be appropriate for the fits. As future work, we note that it would be worthwhile to investigate this on many more data sets, and to observe more carefully how much we gain in accuracy improvement and how much we lose in storage space requirements by choosing additional knee points per feature.

V. RESULTS

In this section, the results of the evaluation described in Section IV will be presented and discussed. As mentioned in Section III-A, the choice of BIN plays an important role in the descriptiveness of FPFH. Many people use FPFH implementations of PCL and Open3D libraries, where the default setting is currently BIN $=$ 11.[1] In our previous work [53], we investigated what happens to the descriptiveness when a smaller number is used. However, to the best of our knowledge, there is no publication that has investigated the same for BIN $>$ 11, so we fill this gap in this section. Another important study is to compare the use of modified features with the original ones.

A. DESCRIPTIVENESS

Figure 10 shows the *precision-recall* curves for different hand-crafted 3D point descriptors on the 7-Scenes RGB-D redkitchen dataset. As mentioned before, we generated the curves by varying the value of τ, between 0.7 and 1.0, with 0.1 steps. For comparison, we selected well-known hand-crafted descriptors (SHOT [29], Spin Images [26], 3DHopD [36]), and we used the same neighborhood radius (0.06) for all methods for fair comparison. Based on our evaluation, the performance of the Spin Images and 3DHopD

[1]See PCL 1.13.1 and Open3D 0.17 versions.

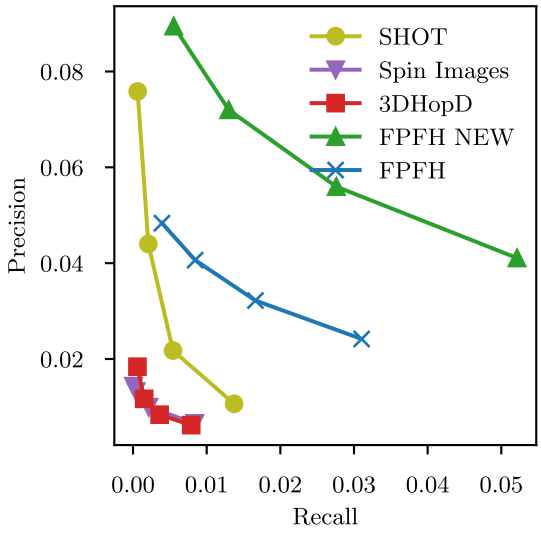


FIGURE 10. Precision-recall curves of different point descriptors.

methods is significantly lower than SHOT and FPFH. The *precision* of the SHOT descriptor is higher than the *precision* of FPFH. The opposite is true for *recall*. The figure also shows the difference between the original and the modified FPFH. It is clear that the modified FPFH is better than the other methods in both *precision* and *recall*.

The Area Under Curve (AUC) values of the corresponding precision-recall curves for the redkitchen dataset are shown in Figure 11. It can be seen that the descriptiveness of the original FPFH with the blue color is inferior to that of the versions using modified features. Furthermore, for this dataset, we have also investigated the effect of the swap of roles between base point and neighbor point mentioned in Section III-B. The orange columns represent the role-swapped version and the green columns represent the no-role-swapped version. It can be concluded that the swap of roles is not justified under the original condition, and in fact worsens the descriptiveness. It can also be seen from the green colored bars that the strongest descriptiveness is at BIN $=$ 27 and starts to deteriorate after that. (This may be because, as is generally true, histograms become undersmoothed above a bin number.) The dashed lines show the descriptors of the PARAMS representations (red for the role-swapped version, cyan for the no-role-swapped version), where we chose one point per feature, which split the empirical CDFs into two intervals. The number of parameters to be stored is 5 per feature in this case, i.e. a total of 15 numbers to be stored. This shows better descriptiveness than in the case BIN $=$ 11, where a total of 33 numbers need to be stored. See Figure 12 for the results (AUC values) of the evaluation on the Redwood livingroom dataset. Here again, we see that the descriptiveness of the original FPFH in blue is inferior to that of the no-role-swapped version using modified features in green. The strongest descriptiveness for this dataset is at BIN $=$ 31 (for the green colored ones) and after that a stagnation is observed (undersmoothing of

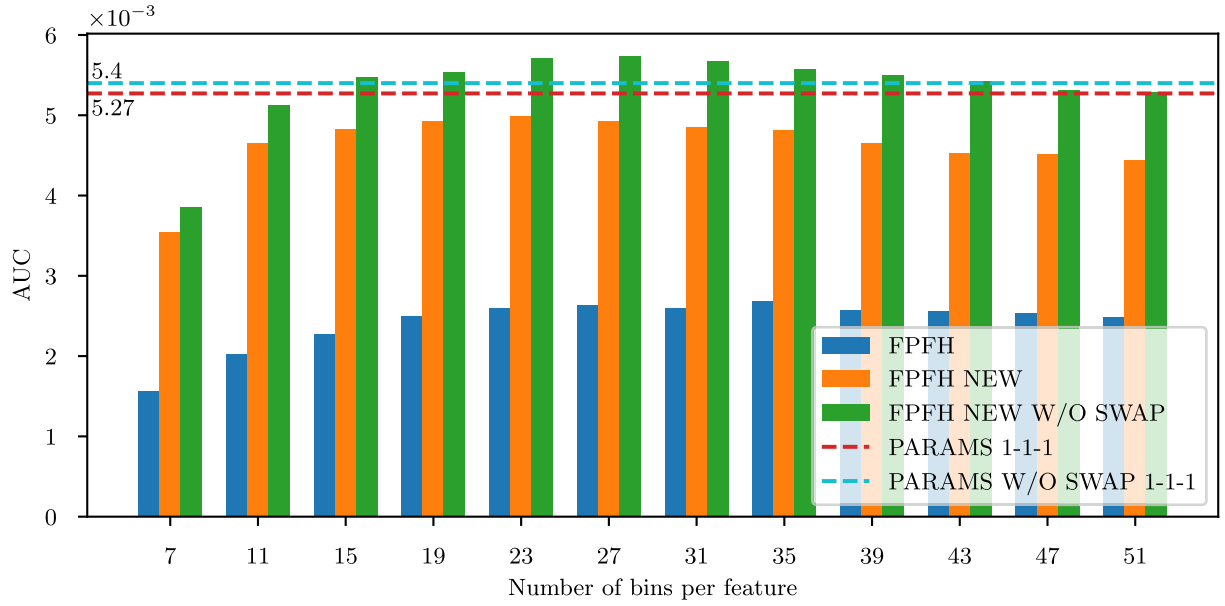


FIGURE 11. AUC of original FPFH, FPFH with modified features and PFDENS with and without swap on the "redkitchen" dataset.

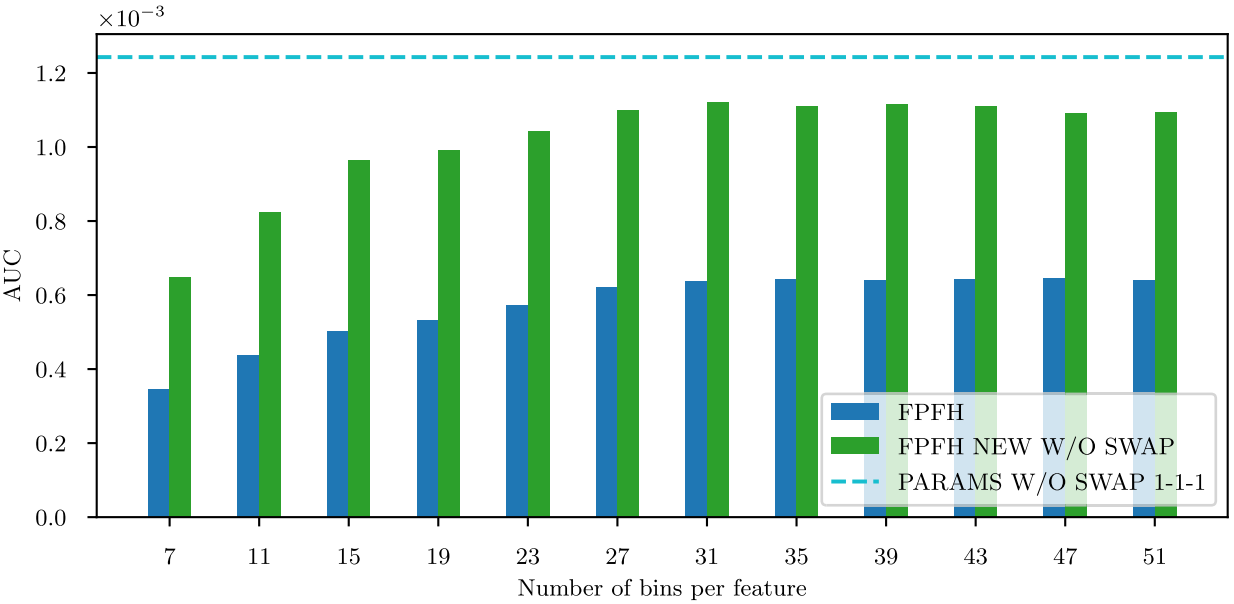


FIGURE 12. AUC of original FPFH, FPFH with modified features and PFDENS without swap on the "livingroom" dataset.

histograms would only occur for even higher BIN numbers). Thus, the maximum is at roughly similar locations for the two data sets. The dashed line here also shows the descriptiveness of the PARAMS representation, where one point per feature was chosen as for the other dataset. This now shows even better descriptiveness than the BIN $= 31$ case. This can occur because in this case the fitted functions can smooth out the noises in the FPFH histograms, which shows the strength of the method. It should be noted that for this dataset, typically lower descriptiveness were obtained. The likely reason for this phenomenon is that there are only a few interesting objects at these "livingroom" scenes, which makes it more difficult to distinguish between surfaces.

To summarize, it can be seen that using the modified features roughly doubles the descriptiveness for both data sets. Furthermore, it is worth choosing a higher BIN number compared to the default. Although, a related question is whether the increase in storage requirements is acceptable and worth it for the given use case. Finally, it can be concluded that the CDF representation of features can be an interesting trade-off between descriptiveness and storage requirements.

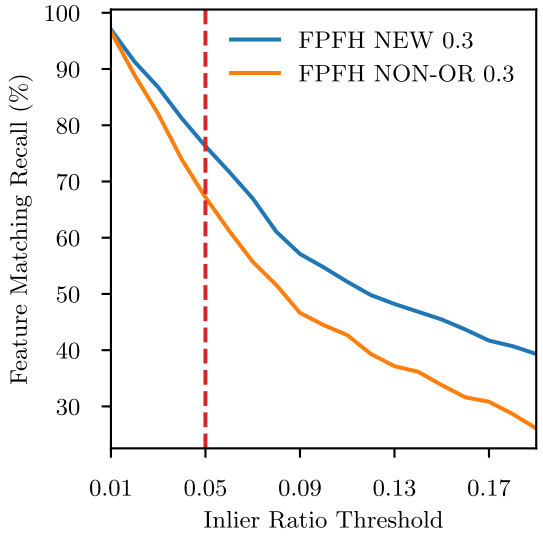


FIGURE 13. Average feature matching recall on the redkitchen dataset under different inlier ratio threshold τ_2.

B. FEATURE MATCHING RECALL

Based on the PPFNet paper [58], the hand-crafted based methods achieved the following results: Spin Images [26] - 0.27, SHOT [29] - 0.21, USC [6] - 0.52, FPFH - 0.36. The parameters of the methods are very important, especially the radius used for querying the neighborhood. Following the authors, we choose the radius parameter for FPFH to be 0.3 also. Using the same evaluation method, the average recall of the modified FPFH is 0.76. It is much better than any other hand-crafted method.

The authors also evaluated learning-based methods: PointNet [61] - 0.76, CGF [11] - 0.72, 3DMatch [39] - 0.85, PPFNet [58] - 0.9. These methods use different radius parameters. Increasing the radius to 0.52, the modified FPFH can achieve 0.83 average recall. This is below the results of the latest learning-based methods, but considering that FPFH does not need learning, we consider it a good result.

We also compared the original FPFH without normal orientation (FPFH NON-OR 0.3) and the modified FPFH (FPFH NEW 0.3), using 0.3 as the neighborhood radius in both cases. Figure 13 shows the average feature matching recall for the two methods under different inlier ratios τ_2. One can see that the modified version has a higher average recall throughout the whole interval. With a slightly stricter threshold, the average recall can go above 80% and could reach even 95%.

C. COMPUTATIONAL COMPLEXITY

The FPFH computation time grows with increasing bin numbers and radius. The parameter fits were performed after the FPFH histograms were defined, for which we therefore chose the cumulative histogram representation of the empirical CDFs where the bin number is 27 (see Section IV-D). The time of parameter fits to an empirical CDF is more influenced by the number of intervals chosen (based on knee points or other points of interest) than by the bin numbers or the radius size. If this method is thought of as a form of compression, the information loss can be controlled by the choice of the number of intervals. We also note that one could make this method independent of the FPFH histogram computation and directly determine parameter fits to empirical CDFs based on the point-pair features, which would yield faster run times. It is beyond the scope of this work to think through and implement the details. Hence, the presentation of the following elapsed times can only be considered as a preliminary result.

Experiments are performed with a single processor core on a virtual machine (Intel Core Processor i5-1135G7 2.4 GHz with 9.12GB RAM) running Ubuntu 22.04.3 LTS operating system. For simplicity, the redkitchen point cloud pair with indices 10 and 11 is selected for the measurements, as their AUC values were always around the mean in the precision-recall evaluations. The neighborhood radius for the point-pair features is now chosen to be 0.3. For these point clouds, the FPFH computtion takes an average of 45.7854 milliseconds per point, followed by the parameter fitting, which takes an average of 39.474 milliseconds per point. The AUC value with FPFH is 0.0123, with parameter fit 0.0098, so 79.58% of the descriptiveness is preserved for this radius, but the storage requirement is reduced to 18.52% (or the compression ratio is 5.4). Thus, in this case, with a significant reduction in storage requirement, there is an acceptable level of information loss with roughly twice the total elapsed time.

VI. CONCLUSION

In this work, we revisited the Fast Point Feature Histogram local feature descriptor. FPFH creates a histogram based on point-pair features calculated for the neighborhood of a base point. We presented three main contributions. First, we modified the point-pair features used by FPFH. As a result, the descriptiveness of the FFPH nearly doubled and became invariant to the inconsistent normal orientation. Second, we evaluated FFPH with different histogram resolution. According to the results, the descriptiveness of the FPFH increases up to 27 bins per feature with both the old and new point-pair features. Finally, we present a method to approximate the CDFs of the three point-pair features. Thus, in some cases, we can reduce the storage space requirement of the feature vectors while maintaining the descriptive ability. For this, we analyzed a special case mathematically (an edge formed by two planes), which we later used in a general case on real point cloud datasets.

As mentioned in Section III-B, the swap of roles between base point and neighbor point should not be done by testing whether condition (1) is satisfied, but by inspecting whether $|n_1 \cdot (p_2 - p_1)|$ is close to 1 and $|n_2 \cdot (p_2 - p_1)|$ is significantly smaller than it. The correct choice of the relevant thresholds should be investigated in future work.

In Section III-C, a realistic, practical generalization of the derivation of CDFs for the meeting of multiple planes is presented, relying on formulas defined for a special case of

two intersecting planes. The generalization is supported by empirical evidence, but future work could explore theoretically what might be behind it (at the moment we only have a mathematical conjecture for the general case). Furthermore, the endpoints of the intervals could be defined in a more sophisticated way, and the increase in descriptiveness and storage requirements when more intervals are chosen could be investigated in more detail.

Finally, it should be mentioned that this work is mainly positioned for theoretical explanations and investigations, thus, for example, it omits the very interesting run-time analysis from a practical point of view, and we could only show preliminary results on this. This would have required a clever implementation that could make the function fitting process as fast as possible. We would like to address this carefully in a future work.

APPENDIX A
FPFH FEATURES WITHOUT ORIENTATION OF NORMAL VECTORS

In this section we will investigate why it will be true that we do not need to perform the costly orientation on the normal vectors, using the modifications that we make to the FPFH features in Section III-B.

Suppose that features are computed for a base point p_1 and its neighbor point p_2, where u and n_2 denote the corresponding normal vectors, as before. We will not orientate the normal vectors before or during the computation of the features, but we will consider one of the orientations relative to the angle subtended by u and n_2 and to the angle subtended by u and the vector $p_2 - p_1$ as the standard. We will examine the features ϕ, θ and α computed for other orientations of the normal vectors with respect to the standard. For the standard orientation, we denote the normal vectors by u^A and n_2^A, respectively, and for the opposite orientation, by u^B and n_2^B, respectively, where $u^B = -u^A$ and $n_2^B = -n_2^A$. Also denote v^A and w^A (v^B and w^B) by the additional vectors of the Darboux frame associated with u^A (u^B). The following are true:

$$\begin{aligned} v^B &= \frac{(p_2 - p_1) \times (-u^A)}{\|(p_2 - p_1) \times (-u^A)\|} \\ &= -\frac{(p_2 - p_1) \times u^A}{\|(p_2 - p_1) \times u^A\|} \\ &= -v^A \end{aligned} \tag{17}$$

$$\begin{aligned} w^B &= (-u^A) \times (-v^A) \\ &= u^A \times v^A \\ &= w^A \end{aligned} \tag{18}$$

For the standard orientation, let the following be true:

$$u \cdot n_2 > 0, \tag{19}$$

$$\frac{u \cdot (p_2 - p_1)}{\|p_2 - p_1\|} \leq 0. \tag{20}$$

The case $u \cdot n_2 = 0$ requires separate analysis, this is left to the reader.

Referring back to Figure 2, this will be the case of the blue color-coded point cloud, if we take the base point p assigned there as p_1. This is because the inequality (19) is equivalent to the case where the angle subtended by u and n_2 falls in the interval $[0, \pi/2)$ or $(3\pi/2, 2\pi]$ (for the normal vectors of the blue color-coded point cloud, the angle is clearly in the interval $[0, \pi/2)$, so it holds). On the other hand, the inequality (20) is equivalent to the case where the angle subtended by u with the vector $p_2 - p_1$ falls in the interval $[\pi/2, 3\pi/2]$ (which also holds in the blue case).

There are four types of orientation cases:

Case I:	u^A and n_2^A,
Case II:	u^B and n_2^B,
Case III:	u^A and n_2^B,
Case IV:	u^B and n_2^A.

Let the values ϕ, θ and α computed for Case I be denoted by $\phi^{A,A}$, $\theta^{A,A}$ and $\alpha^{A,A}$, which are thus computed for the standard orientation, i.e. $u^A \cdot n_2^A > 0$ and

$$\phi^{A,A} = \frac{u^A \cdot (p_2 - p_1)}{\|p_2 - p_1\|} \leq 0.$$

Consider Case II, where the corresponding feature values are denoted by $\phi^{B,B}$ and $\theta^{B,B}$. On the one hand, it is true that

$$\phi^{B,B} = \frac{u^B \cdot (p_2 - p_1)}{\|p_2 - p_1\|} = \frac{-u^A \cdot (p_2 - p_1)}{\|p_2 - p_1\|} = -\phi^{A,A} \tag{21}$$

and hence $\phi^{B,B} \geq 0$. On the other hand, due to

$$u^B \cdot n_2^B = (-u^A) \cdot (-n_2^A) = u^A \cdot n_2^A > 0$$

and $w^B \cdot n_2^B = -w^A \cdot n_2^A$ (using equation (18)), it is true that

$$\begin{aligned} \theta^{B,B} &= \arctan(w^B \cdot n_2^B, u^B \cdot n_2^B) \\ &= \arctan\left(\frac{w^B \cdot n_2^B}{u^B \cdot n_2^B}\right) \\ &= \arctan\left(\frac{w^A \cdot (-n_2^A)}{u^A \cdot n_2^A}\right) \\ &= -\theta^{A,A}. \end{aligned} \tag{22}$$

Furthermore, it is true that

$$\alpha^{B,B} = v^B \cdot n_2^B = (-v^A) \cdot (-n_2^A) = \alpha^{A,A}.$$

Before going on to examine the remaining two cases, let us assume that $w \cdot n_2^A > 0$ is satisfied (the case $w \cdot n_2^A \leq 0$ should be seen in a similar way). Because of this and $u^A \cdot n_2^A > 0$, it is true that

$$0 < \theta^{A,A} < \pi/2. \tag{23}$$

Consider Case III, where the corresponding feature values are denoted by $\phi^{A,B}$ and $\theta^{A,B}$. It is easy to see that

$\phi^{A,B} = \phi^{A,A} \leq 0$. Furthermore, due to $u^A \cdot n_2^B = -u^A \cdot n_2^A < 0$ and $w^A \cdot n_2^B = -w^A \cdot n_2^A < 0$, it is true that

$$\begin{aligned}\theta^{A,B} &= \arctan(w^A \cdot n_2^B, u^A \cdot n_2^B)\\ &= \arctan(-w^A \cdot n_2^A, -u^A \cdot n_2^A)\\ &= \arctan\left(\frac{-w^A \cdot n_2^A}{-u^A \cdot n_2^A}\right) - \pi\\ &= \theta^{A,A} - \pi.\end{aligned}$$

Furthermore, it is true that

$$\alpha^{A,B} = v^A \cdot n_2^B = v^A \cdot (-n_2^A) = -\alpha^{A,A}.$$

Finally, consider Case IV, where the corresponding feature values are denoted by $\phi^{B,A}$ and $\theta^{B,A}$. As in equation (21), it is true that $\phi^{A,B} = -\phi^{A,A} \geq 0$. Furthermore, due to $u^B \cdot n_2^A = -u^A \cdot n_2^A < 0$ and $w^B \cdot n_2^A = w^A \cdot n_2^A > 0$ (using equation (18)), the following is true

$$\begin{aligned}\theta^{B,A} &= \arctan(w^B \cdot n_2^A, u^B \cdot n_2^A)\\ &= \arctan(w^A \cdot n_2^A, -u^A \cdot n_2^A)\\ &= \arctan\left(\frac{w^A \cdot n_2^A}{-u^A \cdot n_2^A}\right) + \pi\\ &= -\theta^{A,A} + \pi.\end{aligned} \tag{24}$$

Furthermore, it is true that

$$\alpha^{B,A} = v^B \cdot n_2^A = (-v^A) \cdot n_2^A = -\alpha^{A,A}.$$

Keeping the assumptions made so far, we summarize our results in Table 1. In each case, we take the following extra steps from Section III-B, recalling equations (7), (8), (9) and (10). In Case II and Case IV, the sign of ϕ is changed according to equation (7) (i.e. it becomes $\phi^{A,A}$), and in Case III its value is not changed (i.e. it remains $\phi^{A,A}$). So in each case we get the same for ϕ as for Case I. For Case II, based on equation (8), what changes in equation (22) is that we get $\theta^{A,A}$ at the end. For Case III, the inequality (23) gives $\theta^{A,B} < -\pi/2$. Using this and equation (9), we get $\theta^{A,A}$. For Case IV, based on equation (8), what changes in equation (24) is that we get $\theta^{A,A} - \pi$ at the end, from now on the same as for Case III. So in each case we get the same for θ as for Case I. Since $u \cdot n_2$ changes sign in exactly the same way as $v \cdot n_2$ for different cases, in Case III and Case IV, the sign of α is changed according to equation (10) (i.e. it becomes $\alpha^{A,A}$), and in Case II its value is not changed (i.e. it remains $\alpha^{A,A}$). This allow us to see that with the extra steps in Section III-B, the orientation of the normal vectors becomes uninteresting.

TABLE 1. The ϕ, θ and α values for the four types of orientation cases.

Cases	ϕ	θ	α
Case I	$\phi^{A,A} \leq 0$	$\theta^{A,A}$	$\alpha^{A,A}$
Case II	$\phi^{B,B} = -\phi^{A,A}$	$\theta^{B,B} = -\theta^{A,A}$	$\alpha^{B,B} = \alpha^{A,A}$
Case III	$\phi^{A,B} = \phi^{A,A}$	$\theta^{A,B} = \theta^{A,A} - \pi$	$\alpha^{A,B} = -\alpha^{A,A}$
Case IV	$\phi^{B,A} = -\phi^{A,A}$	$\theta^{B,A} = -\theta^{A,A} + \pi$	$\alpha^{B,A} = -\alpha^{A,A}$

APPENDIX B
CDFs OF FPFH FEATURES ON THE SLOPE

In the remainder of this section, we assume that there are two intersecting planes whose dihedral angle is $\pi/2 - \gamma$ where $0 \leq \gamma < \pi/2$ (so these two planes form a slope). See Figure 14. The point p_1 is located at a distance ℓ from the intersection of the planes. We take the neighborhood of point p_1, marked in red in the figure. For simplicity, the radius R of the neighborhood is chosen so that it does not hang off the slope in either direction. Furthermore, we assume that the point cloud elements are perfectly aligned on the planes, so there is no noise, and the points are evenly distributed on each part of the planes. (At first glance, these seem to be strong assumptions, but empirically the main findings here will be more or less valid in a more general case.) Without loss of generality, we can choose the axes as shown in the figure, it will be useful, as will be shown later. It is important to note that the features in this section will be examined by taking the neighbor points not from the plane of point p_1, but from the other plane (marked in green in the figure). In the figures, the neighbor point is p_2 and $\rho = \|p_2 - p_1\|$ is the distance between p_1 and p_2. ω (β) denotes the angle subtended by $p_2 - p_1$ (w) and the x-axis. It is easy to see based on the previous comment that $\arcsin(\ell/R) \leq \omega, \beta \leq \pi - \arcsin(\ell/R)$. The translation of the normal vector n_2 to the origin is denoted by n_2'. The line segment of length $\cos(\gamma)$ ($\sin(\gamma)$) is the (scalar) projection of the normal vector n_2' onto the y-axis (z-axis).

Theorem 1 shows the relationships between FPFH features and different angles, the significance of which will be clear below, but briefly it is meant to help us in defining or estimating the distribution functions of FPFH features in the slope case.

Theorem 1: The following are true:

$$\begin{aligned}\theta &= \arctan(\cos(\gamma)\sin(\beta), \sin(\gamma)) \qquad (25)\\ &= \begin{cases}\arctan(\sin(\beta)/\tan(\gamma)) & \text{if } 0 < \gamma < \pi/2, \pi/2 \quad \text{if } \gamma = 0\end{cases}\\ \alpha &= -\cos(\gamma)\cos(\beta) \qquad (26)\\ \phi &= \cos(\gamma)\left(\frac{\ell\sin(\gamma)}{\rho} - \sqrt{\sin^2(\omega) - \frac{\ell^2\cos^2(\gamma)}{\rho^2}}\right). \qquad (27)\end{aligned}$$

Proof: Throughout the proof, we will refer to the notations on the auxiliary figures. It is definitely worth keeping in mind Figure 14, where one can get an overview of the most important vectors and angles.

From now on we make the assumptions: $\omega, \beta \leq \pi/2$ and $\gamma > 0$ (this case is shown in the auxiliary figures), the other cases are left to the reader, as it is very similar (using $\cos(\beta - \pi/2) = \cos(\pi/2 - \beta)$, $\sin(\pi - \omega) = \sin(\omega)$ and if $\gamma = 0$, then n_2 is parallel to the y-axis).

Let us first prove equation (25) for the case $\beta < \pi/2$ and later return to the case $\beta = \pi/2$. To see this, consider the triangle in the left panel of Figure 15. Using the Law of

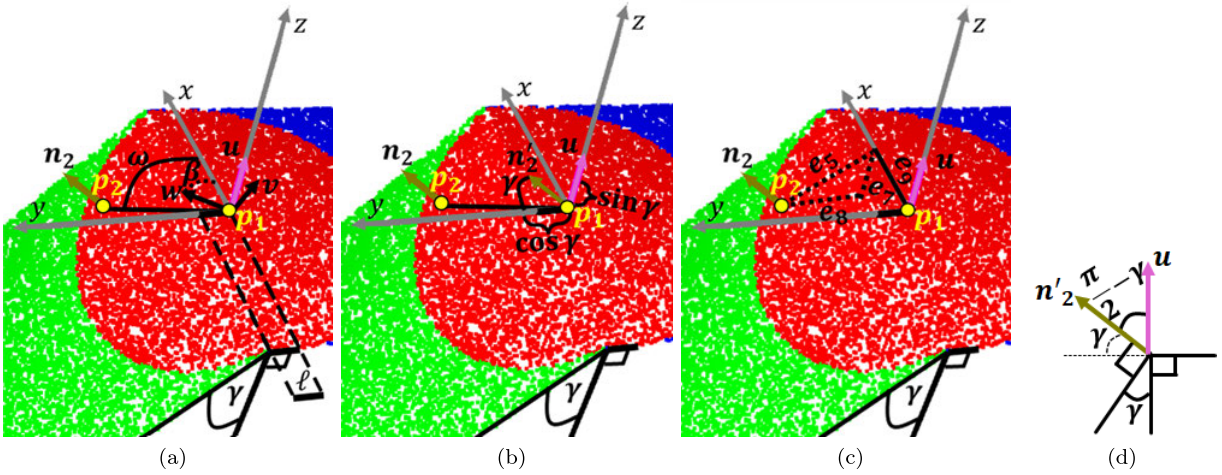


FIGURE 14. The most important vectors, edges and angles needed to prove Theorem 1 and Theorem 3. Further explanation is given in the text.

Cosines, we can express e_1 in the figure:

$$e_1 = \sqrt{\cos^2(\gamma) + \|w\|^2 - 2\,\|w\|\cos(\gamma)\cos(\pi/2-\beta)}$$
$$= \sqrt{\cos^2(\gamma) + 1 - 2\cos(\gamma)\sin(\beta)}.$$

Next, we express e_2 in the middle panel of Figure 15 based on the Pythagorean theorem:

$$e_2 = \sqrt{\sin^2(\gamma) + \cos^2(\gamma) + 1 - 2\cos(\gamma)\sin(\beta)}$$
$$= \sqrt{2 - 2\cos(\gamma)\sin(\beta)}.$$

Then applying the Law of Cosines again for the right panel of Figure 15, we get $\cos(\eta)$, which is the cosine of the angle subtended by w and n'_2:

$$\cos(\eta) = \frac{\|n'_2\|^2 + \|w\|^2 - 2 + 2\cos(\gamma)\sin(\beta)}{2\,\|n'_2\|\,\|w\|}$$
$$= \cos(\gamma)\sin(\beta).$$

It is easy to see that in the case $\beta = \pi/2$, $\eta = \gamma$ (since the vector w is then on the y-axis), so the result for $\cos(\eta)$ is still true. Therefore, the dot product of w and n_2 is $\cos(\gamma)\sin(\beta)$. Based on Figure 14d,

$$u \cdot n_2 = \|u\|\,\|n_2\|\cos(\pi/2-\gamma) = \sin(\gamma).$$

It follows that equation (25) holds.

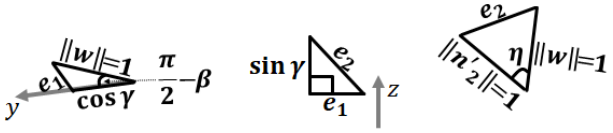


FIGURE 15. The triangles needed for the derivation of equation (25). Further explanation is given in the text.

In the same manner, we can prove, as we saw for the scalar product of w and n_2, that equation (26) holds. To do this,

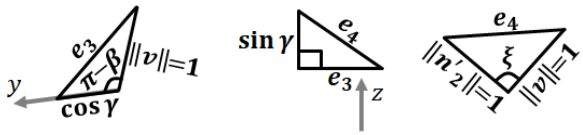


FIGURE 16. The triangles needed for the derivation of equation (26). Further explanation is given in the text.

consider Figure 16, which shows that

$$v \cdot n_2 = \|v\|\,\|n_2\|\cos(\xi) = -\cos(\gamma)\cos(\beta)$$

using the Law of Cosines and the Pythagorean theorem. (For this part, no distinction is made between $\beta = \pi/2$ and $\beta < \pi/2$.)

Finally, we show that equation (27) also holds. To do this, we first consider the triangle in Figure 17a for the case $\omega < \pi/2$, where $e_5 = \sin(\omega)\rho$, and in the special case of $\omega = \pi/2$, e_5 equals ρ (because the ρ is on the yz-plane). We have created another auxiliary figure, Figure 14c, which is particularly useful for the proof of (27), as it clearly shows the relationships between the most important edges (the figure shows that the coordinates of point p_2 are (e_9, e_8, e_7)). In Figure 17b, there are several triangles connected to each other. Take the one with sides e_5, e_6, ℓ and use the Law of Cosines:

$$\sin^2(\omega)\rho^2 = e_6^2 + \ell^2 - 2e_6\ell\cos\left(\frac{\pi}{2}+\gamma\right)$$

then express the two possible solutions for e_6 using the quadratic formula:

$$e_6 = \ell\cos\left(\frac{\pi}{2}+\gamma\right) \pm \sqrt{\sin^2(\omega)\rho^2 + \ell^2\cos^2\left(\frac{\pi}{2}+\gamma\right) - \ell^2}. \tag{28}$$

Now consider a triangle with sides e_5, e_7 and e_8, where e_5 is the hypotenuse, so it is true that $e_8 < e_5$. On the other hand,

$\ell \leq e_8$, which implies that $\ell < \sin(\omega)\rho$ holds. Using this, the following can be derived:

$$\begin{aligned}\ell^2 &< \sin^2(\omega)\rho^2\\ \ell^2\cos^2\left(\frac{\pi}{2}+\gamma\right) &< \sin^2(\omega)\rho^2 + \ell^2\cos^2\left(\frac{\pi}{2}+\gamma\right) - \ell^2\\ \ell\cos\left(\frac{\pi}{2}+\gamma\right) &< \sqrt{\sin^2(\omega)\rho^2 + \ell^2\cos^2\left(\frac{\pi}{2}+\gamma\right) - \ell^2}.\end{aligned}$$

Using this, and since e_6 must be positive, we get that in (28) the plus-minus symbol ($\pm$) can actually be replaced by "+":

$$\begin{aligned}e_6 &= \ell\cos\left(\frac{\pi}{2}+\gamma\right) + \sqrt{\sin^2(\omega)\rho^2 + \ell^2\cos^2\left(\frac{\pi}{2}+\gamma\right) - \ell^2}\\ &= \ell\cos\left(\frac{\pi}{2}+\gamma\right) + \sqrt{\sin^2(\omega)\rho^2 - \ell^2\sin^2\left(\frac{\pi}{2}+\gamma\right)}\\ &= -\ell\sin(\gamma) + \sqrt{\sin^2(\omega)\rho^2 - \ell^2\cos^2(\gamma)}.\end{aligned} \tag{29}$$

Examining the right triangle with the γ angle, we get $e_7 = \cos(\gamma)e_6$. Finally, consider the triangle in Figure 17c, which shows: $\cos(\pi - \lambda) = e_7/\rho = \frac{\cos(\gamma)e_6}{\rho}$. We get $\cos(\lambda) = -\frac{\cos(\gamma)e_6}{\rho}$, which is the cosine of the angle subtended by u and $p_2 - p_1$. We have now deduced equation (27) and the proof is complete. □

Let us now state the corollary of Theorem 1, which is easy to see using the fact that $\arcsin(\ell/R) \leq \omega, \beta \leq \pi - \arcsin(\ell/R)$. From now on, ϕ_0 denotes the minimum value of ϕ that is equal to

$$\phi_0 = \cos(\gamma)\left(\frac{\ell\sin(\gamma)}{R} - \sqrt{1 - \frac{\ell^2\sin^2(\gamma)}{R^2}}\right) \tag{30}$$

(ϕ takes its minimum at $\omega = \pi/2$ and $\rho = R$).

Corollary 1: The following are true:

$$\begin{aligned}&\arctan(\cot(\gamma)\ell/R) \leq \theta \leq \arctan(\cot(\gamma)) \text{ (if } \gamma > 0)\\ &-\sqrt{1-\frac{\ell^2}{R^2}}\cos(\gamma) \leq \alpha \leq \sqrt{1-\frac{\ell^2}{R^2}}\cos(\gamma)\\ &\phi_0 \leq \phi \leq 0.\end{aligned}$$

The following proposition establishes a relationship between the CDF of β, denoted by F_β, and the CDFs of the features θ (F_θ) and α (F_α).

Proposition 1: For the CDF of θ, it is true that

$$F_\theta(t) = \begin{cases}0 & \text{for } t < \pi/2\\ 1 & \text{for } t \geq \pi/2\end{cases} \tag{31}$$

if $\gamma = 0$, otherwise

$$F_\theta(t) = \begin{cases}0 & \text{for } t < \arctan(\cot(\gamma)\ell/R)\\ 2\cdot F_\beta(\arcsin(\tan(t)\tan(\gamma))) & \\ \quad \text{for } \arctan(\cot(\gamma)\ell/R) \leq t \leq \arctan(\cot(\gamma)) & \\ 1 & \text{for } t > \arctan(\cot(\gamma))\end{cases} \tag{32}$$

For the CDF of α, it is true that

$$F_\alpha(t) = \begin{cases}0 & \text{for } t < -\sqrt{1-\frac{\ell^2}{R^2}}\cos(\gamma)\\ F_\beta(\arccos(-t/\cos(\gamma))) & \\ \quad \text{for } -\sqrt{1-\frac{\ell^2}{R^2}}\cos(\gamma) \leq t \leq \sqrt{1-\frac{\ell^2}{R^2}}\cos(\gamma) & \\ 1 & \text{for } t > \sqrt{1-\frac{\ell^2}{R^2}}\cos(\gamma)\end{cases} \tag{33}$$

Proof: To prove this, we will use equations (25) and (26) in Theorem 1 and Corollary 1. Equation (31) is easy to check. To prove equation (32), use the fact that if $\sin(\beta) \leq \tan(t)\tan(\gamma)$, then $\sin(\pi - \beta) \leq \tan(t)\tan(\gamma)$ and vice versa. This is why the multiplication by 2 appears in the derivation below:

$$\begin{aligned}F_\theta(t) &= \mathbb{P}(\theta \leq t) = \mathbb{P}(\arctan(\sin(\beta)/\tan(\gamma)) \leq t)\\ &= \mathbb{P}(\sin(\beta) \leq \tan(t)\tan(\gamma))\\ &= 2\cdot\mathbb{P}(\beta \leq \arcsin(\tan(t)\tan(\gamma)))\\ &= 2\cdot F_\beta(\arcsin(\tan(t)\tan(\gamma)))\end{aligned}$$

where $\arctan(\cot(\gamma)\ell/R) \leq t \leq \arctan(\cot(\gamma))$ (the other cases are straightforward). We show that equation (33) holds:

$$\begin{aligned}F_\alpha(t) &= \mathbb{P}(\alpha \leq t) = \mathbb{P}(-\cos(\gamma)\cos(\beta) \leq t)\\ &= \mathbb{P}(\cos(\beta) \geq -t/\cos(\gamma))\\ &= \mathbb{P}(\beta \leq \arccos(-t/\cos(\gamma)))\\ &= F_\beta(\arccos(-t/\cos(\gamma)))\end{aligned}$$

where $-\sqrt{1-\frac{\ell^2}{R^2}}\cos(\gamma) \leq t \leq \sqrt{1-\frac{\ell^2}{R^2}}\cos(\gamma)$ (the other cases are straightforward). □

The function F_β plays a central role in determining the functions F_θ and F_α according to Proposition 1. We will not give an explicit formula for this function F_β, but the following proposition gives a lower and an upper estimates of the function using the CDF of ω, denoted by F_ω.

Proposition 2: For the CDF of β, it is true:

$$F_\beta(t) \leq \begin{cases}F_\omega(\arctan(\tan(t)/\sin(\gamma))) & \\ \quad \text{for } \arcsin(\ell/R) \leq t < \pi/2 & \\ F_\omega(t) \quad \text{for } \pi/2 < t \leq \pi - \arcsin(\ell/R)\end{cases} \tag{34}$$

$$F_\beta(t) \geq \begin{cases}F_\omega(t) \quad \text{for } \arcsin(\ell/R) \leq t < \pi/2 & \\ F_\omega(\pi + \arctan(\tan(t)/\sin(\gamma))) & \\ \quad \text{for } \pi/2 < t \leq \pi - \arcsin(\ell/R)\end{cases} \tag{35}$$

$$F_\beta(\pi/2) = 1/2 = F_\omega(\pi/2) \tag{36}$$

Proof: Equation (36) is easy to see (in a similar way to equation (51)). Now, let us assume that

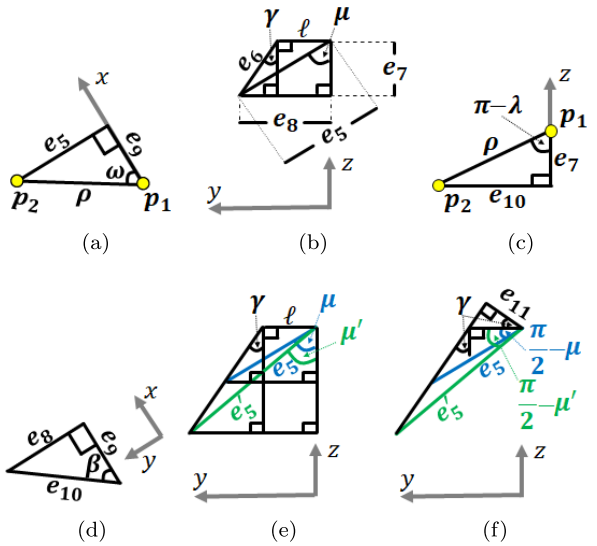


FIGURE 17. The triangles needed to prove Theorem 1, Theorem 2, Proposition 2 and Theorem 3. Further explanation is given in the text.

$\arcsin(\ell/R) \leq t < \pi/2$ and consider the triangle in Figure 17a, which shows the following relationships:

$$e_5 = \rho \sin(\omega) \tag{37}$$
$$\cos(\omega) = e_9/\rho. \tag{38}$$

Figure 17d shows that

$$\tan(\beta) = e_8/e_9. \tag{39}$$

From Figure 17b, take the triangle with sides e_5, e_7, e_8 where it is true that

$$e_8^2 = e_5^2 - e_7^2 \tag{40}$$

thus $e_8 < e_5$, and using equations (37), (38) and (39):

$$\begin{aligned} e_8 &< e_5 \\ e_8 &< \rho \sin(\omega) \\ e_8/e_9 &< \rho \sin(\omega)/e_9 \\ \tan(\beta) &< \tan(\omega) \\ \beta &< \omega. \end{aligned} \tag{41}$$

Examining the right triangle with the angle γ in Figure 17b, we obtain that

$$\sin(\gamma) = (e_8 - \ell)/e_6 \tag{42}$$
$$e_6^2 = e_7^2 + (e_8 - \ell)^2. \tag{43}$$

Using equations (37), (38), (39), (40), (42) and (43) and the fact that $e_5 > e_6$,

$$\begin{aligned} e_8^2 e_6^2 &= \left(e_5^2 - e_7^2\right)\left(e_7^2 + (e_8 - \ell)^2\right) \\ e_8^2 e_6^2 &= e_5^2(e_8 - \ell)^2 + e_7^2\left(e_5^2 - \left((e_8 - \ell)^2 + e_7^2\right)\right) \\ e_8^2 e_6^2 &= e_5^2(e_8 - \ell)^2 + e_7^2(e_5^2 - e_6^2) \\ e_8^2 e_6^2 &> e_5^2(e_8 - \ell)^2 \\ e_8 e_6 &> \rho \sin(\omega)(e_8 - \ell) \\ \frac{e_8}{e_9} &> \frac{\rho \sin(\omega)(e_8 - \ell)}{e_9 e_6} = \frac{\rho \sin(\omega)}{e_9} \cdot \frac{e_8 - \ell}{e_6} \\ \tan(\beta) &> \tan(\omega)\sin(\gamma). \end{aligned} \tag{44}$$

For $\arcsin(\ell/R) \leq t < \pi/2$, if $\tan(\beta) \leq \tan(t)$, then $\tan(\omega)\sin(\gamma) \leq \tan(t)$ also holds based on equation (44), so the event $\tan(\omega)\sin(\gamma) \leq \tan(t)$ is at least as frequent as $\tan(\beta) \leq \tan(t)$, therefore

$$\begin{aligned} F_\beta(t) &= \mathbb{P}(\beta \leq t) = \mathbb{P}(\tan(\beta) \leq \tan(t)) \\ F_\beta(t) &\leq \mathbb{P}(\tan(\omega)\sin(\gamma) \leq \tan(t)) \\ F_\beta(t) &\leq \mathbb{P}(\omega \leq \arctan(\tan(t)/\sin(\gamma))) \\ F_\beta(t) &\leq F_\omega(\arctan(\tan(t)/\sin(\gamma))). \end{aligned}$$

If $\omega \leq t$, then $\beta \leq t$ also holds based on equation (41), as before, we get the following:

$$F_\beta(t) = \mathbb{P}(\beta \leq t) \geq \mathbb{P}(\omega \leq t) = F_\omega(t).$$

The proof for $\pi/2 < t \leq \pi - \arcsin(\ell/R)$ is similar and the details are left to the reader. □

The following theorem gives an upper estimate of the CDF of feature ϕ (F_ϕ), using the function F_ω.

Theorem 2: For the CDF of ϕ, it is true that

$$F_\phi(t) \leq 1 - F_\omega(\arcsin(t/\phi_0)) \tag{45}$$

where $\phi_0 \leq t \leq 0$.

Proof: Let us first consider (again) the triangle in Figure 17a for the case $\omega < \pi/2$, where $\cos^2(\omega) = e_9^2/\rho^2$. Based on the right panel of the same figure, it is true that $\sin^2(\pi - \lambda) = \sin^2(\lambda) = e_{10}^2/\rho^2$. Now, consider the triangle in Figure 17d, where $e_{10}^2 = e_8^2 + e_9^2$. Combining the above, it follows that

$$\begin{aligned} \sin^2(\lambda) &= \frac{e_{10}^2}{\rho^2} = \frac{e_8^2}{\rho^2} + \frac{e_9^2}{\rho^2} = \frac{e_8^2}{\rho^2} + \cos^2(\omega) \\ 1 - \cos^2(\lambda) &= \frac{e_8^2}{\rho^2} + 1 - \sin^2(\omega) \\ \phi = \cos(\lambda) &= \pm\sqrt{\sin^2(\omega) - \frac{e_8^2}{\rho^2}} = -\sqrt{\sin^2(\omega) - \frac{e_8^2}{\rho^2}} \end{aligned} \tag{46}$$

where the plus-minus symbol ($\pm$) can actually be replaced by "$-$" in the last equality because of $\phi \leq 0$. In the special case $\omega = \pi/2$, e_{10} equals e_8 (because the e_{10} is on the yz-plane), from now on, e_{10} is denoted by e_0 for this special case. Similar reasoning applies to this case, and for $\rho = R$, we can see that

$$\phi_0 = -\sqrt{1 - \frac{e_0^2}{R^2}}. \tag{47}$$

For fixed γ and ℓ, the larger e_5, the smaller μ will be, as shown in Figure 17e. To see this correctly, consider Figure 17f and assume $e_5 < e_5'$:

$$\begin{aligned} \cos(\gamma + \pi/2 - \mu) &= e_{11}/e_5 \\ \cos(\gamma + \pi/2 - \mu') &= e_{11}/e_5'. \end{aligned}$$

Using this and the fact that

$$0 \leq \gamma + \pi/2 - \mu, \gamma + \pi/2 - \mu' \leq \pi/2$$

because $\gamma \leq \mu, \mu'$ holds (based on $\tan(\gamma) = (e_8 - \ell)/e_7 \leq e_8/e_7 = \tan(\mu)$), we get

$$\begin{aligned}
\cos(\gamma + \pi/2 - \mu) &= (e_5'/e_5) \cdot \cos(\gamma + \pi/2 - \mu') \\
\cos(\gamma + \pi/2 - \mu) &> \cos(\gamma + \pi/2 - \mu') \\
\gamma + \pi/2 - \mu &< \gamma + \pi/2 - \mu' \\
\mu' &< \mu.
\end{aligned}$$

For $\omega = \pi/2$ and $\rho = R$, e_5 is the largest and then $e_5 = R$ (because $e_5 = \rho \sin(\omega) \leq R \sin(\omega) \leq R$). The angle μ associated with this case will be the smallest, denoted by μ_0. Furthermore, it is easy to see that

$$\sin(\mu) = \frac{e_8}{\rho \sin(\omega)} \tag{48}$$

(based on Figure 17b) and

$$\sin(\mu_0) = \frac{e_0}{R}. \tag{49}$$

From equations (46), (47), (48) and (49), we conclude that

$$\begin{aligned}
\sin(\mu_0) &\leq \sin(\mu) \\
\frac{e_0}{R} &\leq \frac{e_8}{\rho \sin(\omega)} \\
\frac{e_0^2}{R^2} &\leq \frac{e_8^2}{\rho^2 \sin^2(\omega)} \\
\frac{e_0^2}{R^2} \sin^2(\omega) &\leq \frac{e_8^2}{\rho^2} \\
-\frac{e_8^2}{\rho^2} &\leq -\frac{e_0^2}{R^2} \sin^2(\omega) \\
\sin^2(\omega) - \frac{e_8^2}{\rho^2} &\leq \sin^2(\omega)\left(1 - \frac{e_0^2}{R^2}\right) \\
\sqrt{\sin^2(\omega) - \frac{e_8^2}{\rho^2}} &\leq \sin(\omega)\sqrt{1 - \frac{e_0^2}{R^2}} \\
-\sqrt{\sin^2(\omega) - \frac{e_8^2}{\rho^2}} &\geq \sin(\omega)\left(-\sqrt{1 - \frac{e_0^2}{R^2}}\right) \\
\phi &\geq \phi_0 \sin(\omega).
\end{aligned} \tag{50}$$

For $\phi_0 \leq t \leq 0$, if $\phi \leq t$, then $\phi_0 \sin(\omega) \leq t$ also holds based on equation (50), so the event $\phi_0 \sin(\omega) \leq t$ is at least as frequent as $\phi \leq t$, therefore

$$\begin{aligned}
F_\phi(t) = \mathbb{P}(\phi \leq t) &\leq \mathbb{P}(\phi_0 \sin(\omega) \leq t) = \\
&= \mathbb{P}(\sin(\omega) \geq t/\phi_0) = \mathbb{P}(\omega \geq \arcsin(t/\phi_0)) = \\
&= 1 - F_\omega(\arcsin(t/\phi_0))
\end{aligned}$$

which proves the theorem. □

Two questions can be raised about the former result:

- Is there any way to give a lower bound on the function F_ϕ using the function F_ω? Unfortunately, we have not been able to find one so far.
- Could we use the upper estimate to obtain some formula that approximates the function F_ϕ even better? This question will be explored below.

It can be seen from the resulting upper estimate that for $\phi_0 \leq t \leq 0$:

$$1/2 \leq 1 - F_\omega(\arcsin(t/\phi_0)) \leq 1,$$

since $0 \leq \arcsin(t/\phi_0) \leq \pi/2$ and

$$F_\omega(\pi/2) = 1/2, \tag{51}$$

where the latter is shown by the fact that if we take ω to the y-axis (i.e. to $\omega = \pi/2$), then it will be exactly half of the total area of the neighborhood on the slope. (A detailed analysis of function F_ω will be included below.) To obtain an estimate that takes values over the entire interval $[0, 1]$, the argument of the function F_ω should be transformed to take values over the interval $[\arcsin(\ell/R), \pi - \arcsin(\ell/R)]$. The following heuristic will serve this purpose. So we look for the argument transformation in the following form:

$$C_1 \arcsin(t/\phi_0) + C_2 \tag{52}$$

where we want C_1 and C_2 to be chosen such that the minimum and maximum values of t are the two endpoints of the interval $[\arcsin(\ell/R), \pi - \arcsin(\ell/R)]$. Since for $t = 0$ the argument function in (52) will be C_2, choose C_2 as $\arcsin(\ell/R)$. And for $t = \phi_0$ the argument function at (52) will be $C_1 \cdot (\pi/2) + \arcsin(\ell/R)$ (using the previous choice C_2), so choose C_1 such that:

$$\begin{aligned}
C_1 \cdot (\pi/2) + \arcsin(\ell/R) &= \pi - \arcsin(\ell/R) \\
C_1 &= \frac{\pi - 2\arcsin(\ell/R)}{\pi/2} \\
C_1 &= 2\left(1 - \frac{2\arcsin(\ell/R)}{\pi}\right).
\end{aligned}$$

To summarise, we obtain the following estimate (denoted by G_ϕ):

$$F_\phi(t) \approx G_\phi = 1 - F_\omega(C_1 \arcsin(t/\phi_0) + C_2) \tag{53}$$

where $C_1 = 2(1 - 2\arcsin(\ell/R)/\pi)$ and $C_2 = \arcsin(\ell/R)$. So far, our method has only been evaluated by visual inspection, but the results are encouraging, see Figure 22. It can also be seen from the figure that $F_\phi(t) \leq G_\phi(t)$ for $\phi_0 \leq t \leq 0$. (We will return to the other components of the figure later.)

Based on the results obtained so far, it has been possible to use the estimates to derive the CDFs of the FPFH features back to the task of determining the function F_ω. A demonstrative figure has been prepared for this, see Figure 18. This illustrates that if the angle ω goes from the intersection of the planes to t, how much of the resulting area, denoted by S_t, will be part of the area of the complete neighborhood on the slope, which will be $\mathbb{P}(\omega \leq t)$ (assuming no noise and the plane is sampled uniformly). The complete neighborhood on the slope is denoted by S. So the task will be to determine the area of the neighborhood on the slope as

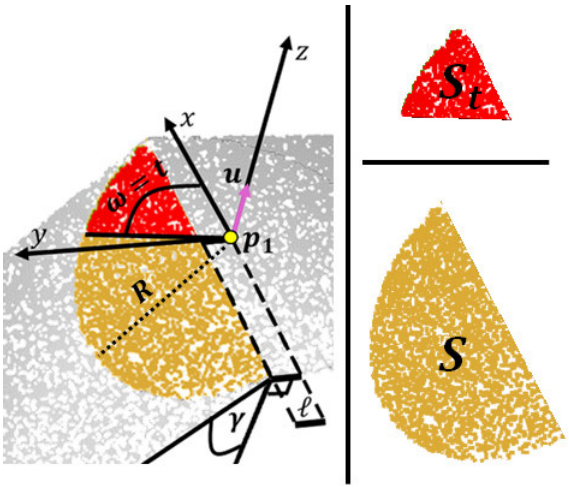


FIGURE 18. Areas defined by the points on the slope: 1) if the angle ω goes from the intersection of the planes to t (surface marked in red, denoted by S_t), 2) if ω goes through the complete neighborhood belonging to the slope (surface marked in orange, denoted by S).

a function of ω, for which Theorem 3 is stated. (Note that for $\ell \neq 0$, this will not simply be the area of a circular sector.)

Theorem 3: For the CDF of ω, it is true that

$$F_\omega(t) = \begin{cases} 0 & \text{for } t < \arcsin(\ell/R) \\ \text{area}(S_t)/\,\text{area}(S) & \\ \quad \text{for } \arcsin(\ell/R) \le t < \pi/2 & \\ 1/2 \quad \text{for } t = \pi/2 & \\ 1 - F_\omega(\pi - t) & \\ \quad \text{for } \pi/2 < t \le \pi - \arcsin(\ell/R) & \\ 1 & \text{for } t > \pi - \arcsin(\ell/R) \end{cases}$$

where the area of S_t (S) is denoted by area(S_t) (area(S)). For the case $\gamma = 0$,

$$\text{area}(S) = (R^2 - \ell^2)(\pi/2)$$

and

$$\text{area}(S_t) = \frac{1}{2} \cdot \left((R^2 - \ell^2)(\pi/2) - R^2 \arcsin\left(\frac{\cos(t)R}{\sqrt{R^2 - \ell^2}}\right) - \frac{\ell^2 \ln\left(\sqrt{R^2\sin^2(t)/\ell^2 - 1} + \sin(t)R/\ell\right)}{\tan(t)} + \ell^2 \arctan\left(\frac{\cos(t)R}{\sqrt{R^2\sin^2(t) - \ell^2}}\right)\right).$$

For the case $\gamma > 0$,

$$\text{area}(S) = \left((R^2 - \ell^2) \arcsin\left(\frac{1}{\sqrt{1 + \frac{\ell^2\sin^2(\gamma)}{R^2 - \ell^2}}}\right) + \ell^2 \sin^2(\gamma) \arctan\left(\frac{\sqrt{R^2 - \ell^2}}{\ell \sin(\gamma)}\right) - \ell \sin(\gamma)\sqrt{R^2 - \ell^2}\right). \quad (54)$$

The area(S_t) is the sum of two terms,

$$\text{area}(S_t) = \text{Int}_1 + \text{Int}_2$$

where

$$\begin{aligned} \text{Int}_1 = \frac{1}{2} \cdot \Bigg(& (R^2 - \ell^2) \arcsin\left(\frac{1}{\sqrt{1 + \frac{\ell^2\sin^2(\gamma)}{R^2 - \ell^2}}}\right) \\ & + \ell^2 \sin^2(\gamma) \arctan\left(\frac{\sqrt{R^2 - \ell^2}}{\ell \sin(\gamma)}\right) \\ & - \ell \sin(\gamma)\sqrt{R^2 - \ell^2} \\ & - \frac{\ell^2 \sin(\gamma)}{\tan^2(t)} \arcsin\left(\frac{1}{\sqrt{1 + \tan^2(t)\sin^2(\gamma)}}\right) \\ & - \ell^2 \sin^2(\gamma) \arctan\left(\frac{1}{\tan(t)\sin(\gamma)}\right) \\ & + \frac{\ell^2 \sin(\gamma)}{\tan(t)}\Bigg), \end{aligned} \quad (55)$$

$$\begin{aligned} \text{Int}_2 = -\frac{1}{2} \cdot \Bigg(& R^2 \arcsin\left(\frac{\cos(t)R}{\sqrt{R^2 - \ell^2\cos^2(\gamma)}}\right) \\ & - \ell^2 \arcsin\left(\frac{\cos(t)}{\sqrt{1 - \cos^2(\gamma)\sin^2(t)}}\right) \\ & + \frac{\ell^2 \cos^2(\gamma)}{\tan(t)} \ln\left(\frac{\sqrt{E_{t,\ell,\gamma,R}} + \sin(t)R}{\ell(1 + \sin(\gamma))}\right) \\ & - \ell^2 \cos^2(\gamma) \arctan\left(\frac{\cos(t)R}{\sqrt{E_{t,\ell,\gamma,R}}}\right) \\ & + \ell^2 \cos^2(\gamma) \arctan\left(\frac{1}{\sqrt{\tan(t)\sin(\gamma)}}\right)\Bigg) \end{aligned} \quad (56)$$

using the notation $E_{t,\ell,\gamma,\rho}$, which abbreviates

$$\rho^2 \sin^2(t) - \ell^2 \cos^2(\gamma).$$

Proof: First we deal with the case $\arcsin(\ell/R) \le t < \pi/2$ and $\gamma > 0$. Our proof starts with observations already alluded to in the proofs of Theorem 1 and Proposition 2. The neighbor point p_2 of point p_1, which is on the slope, has coordinates (e_9, e_8, e_7), see Figure 14c. For better understanding, from now on, the notations e_9, e_8, e_7 will be replaced by X, Y, Z, respectively. Based on equation (38),

$$X = \rho \cos(\omega). \quad (57)$$

Examining the right triangle with the γ angle in Figure 17b and using equation (29), we get

$$\begin{aligned} Y - \ell &= \sin(\gamma) e_6 \\ Y &= \sin(\gamma) e_6 + \ell \\ &= -\ell \sin^2(\gamma) + \ell + \sin(\gamma)\sqrt{\sin^2(\omega)\rho^2 - \ell^2 \cos^2(\gamma)} \\ &= \ell \cos^2(\gamma) + \sin(\gamma)\sqrt{\sin^2(\omega)\rho^2 - \ell^2 \cos^2(\gamma)} \end{aligned} \quad (58)$$

and

$$Z = \frac{Y - \ell}{\tan(\gamma)}. \quad (59)$$

The area of S_t (area(S_t)) can be defined by a surface integral, where Z is a function of Y and the derivative of Z with respect to Y is $1/\tan(\gamma)$ based on equation (59):

$$\begin{aligned} \text{area}(S_t) &= \iint_{S_t} \sqrt{\left(\frac{\partial Z}{\partial X}\right)^2 + \left(\frac{\partial Z}{\partial Y}\right)^2 + 1}\, \mathrm{d}X\, \mathrm{d}Y \\ &= \iint_{S_t} \sqrt{0 + \frac{1}{\tan^2(\gamma)} + \frac{\sin^2(\gamma)}{\sin^2(\gamma)}}\, \mathrm{d}X\, \mathrm{d}Y \\ &= \iint_{S_t} \sqrt{\frac{\cos^2(\gamma) + \sin^2(\gamma)}{\sin^2(\gamma)}}\, \mathrm{d}X\, \mathrm{d}Y \\ &= \iint_{S_t} \frac{1}{\sin(\gamma)}\, \mathrm{d}X\, \mathrm{d}Y \end{aligned}$$

Since we should get the result of the integral as a function of ω, we make transformation from Cartesian coordinates (X, Y) to polar coordinates (ρ, ω) using equations (58) and (59). The Jacobian matrix of the transformation is

$$\begin{bmatrix} -\rho\cos(\omega) & cos(\omega) \\ \frac{\sin(\gamma)\rho^2\cos(\omega)\sin(\omega)}{\sqrt{\sin^2(\omega)\rho^2 - \ell^2\cos^2(\gamma)}} & \frac{\sin(\gamma)\rho\sin^2(\omega)}{\sqrt{\sin^2(\omega)\rho^2 - \ell^2\cos^2(\gamma)}} \end{bmatrix}$$

and the absolute value of the Jacobian determinant is

$$\frac{\rho^2 \sin(\gamma)\sin(\omega)}{\sqrt{\sin^2(\omega)\rho^2 - \ell^2\cos^2(\gamma)}}.$$

To continue integration, we need to apply the boundary conditions. It is easy to see that the lower limit of ω is $\arcsin(\ell/\rho)$ and the upper limit is t. For this reason, it must also be true that $\arcsin(\ell/\rho) \leq t$ and therefore $\ell/\sin(t) \leq \rho$. Clearly, the upper limit of ρ is R. Since $\sin(\gamma)$ is eliminated after simplification, the integral is

$$\begin{aligned} &\int_{\ell/\sin(t)}^{R} \int_{\arcsin(\ell/\rho)}^{t} \frac{\rho^2 \sin(\omega)}{\sqrt{\sin^2(\omega)\rho^2 - \ell^2\cos^2(\gamma)}}\, \mathrm{d}\omega\, \mathrm{d}\rho \\ &= \int_{\ell/\sin(t)}^{R} \rho^2 \left[-\frac{\arcsin\left(\frac{\rho\cos(\omega)}{\sqrt{\rho^2 - \ell^2\cos^2(\gamma)}}\right)}{\rho} \right]_{\arcsin(\ell/\rho)}^{t} \mathrm{d}\rho \end{aligned}$$

The above integral naturally splits into two parts, e.g. $\text{area}(S_t) = \text{Int}_1 + \text{Int}_2$, which will be evaluated separately:

$$\begin{aligned} \text{Int}_1 &= \int_{\ell/\sin(t)}^{R} \rho \arcsin\left(\frac{1}{\sqrt{1 + \frac{\ell^2\sin^2(\gamma)}{\rho^2 - \ell^2}}}\right) \mathrm{d}\rho \\ \text{Int}_2 &= -\int_{\ell/\sin(t)}^{R} \rho \arcsin\left(\frac{\cos(t)}{\sqrt{1 + \frac{\ell^2\cos^2(\gamma)}{\rho^2}}}\right) \mathrm{d}\rho \end{aligned}$$

where we used $\cos(\arcsin(\ell/\rho)) = \sqrt{1 - \ell^2/\rho^2}$ and $-\ell^2\cos^2(\gamma) = -\ell^2 + \ell^2\sin^2(\gamma)$. Let us first compute Int_1,

$$\begin{aligned} \text{Int}_1 = \frac{1}{2} \cdot \Bigg[&(\rho^2 - \ell^2) \arcsin\left(\frac{1}{\sqrt{1 + \frac{\ell^2\sin^2(\gamma)}{\rho^2 - \ell^2}}}\right) \\ &+ \ell^2 \sin^2(\gamma) \arctan\left(\frac{\sqrt{\rho^2 - \ell^2}}{\ell\sin(\gamma)}\right) \\ &- \ell\sin(\gamma)\sqrt{\rho^2 - \ell^2} \Bigg]_{\ell/\sin(t)}^{R}, \end{aligned}$$

evaluating the antiderivative at the endpoints, we get equation (55) where for $\rho = \ell/\sin(t)$, using

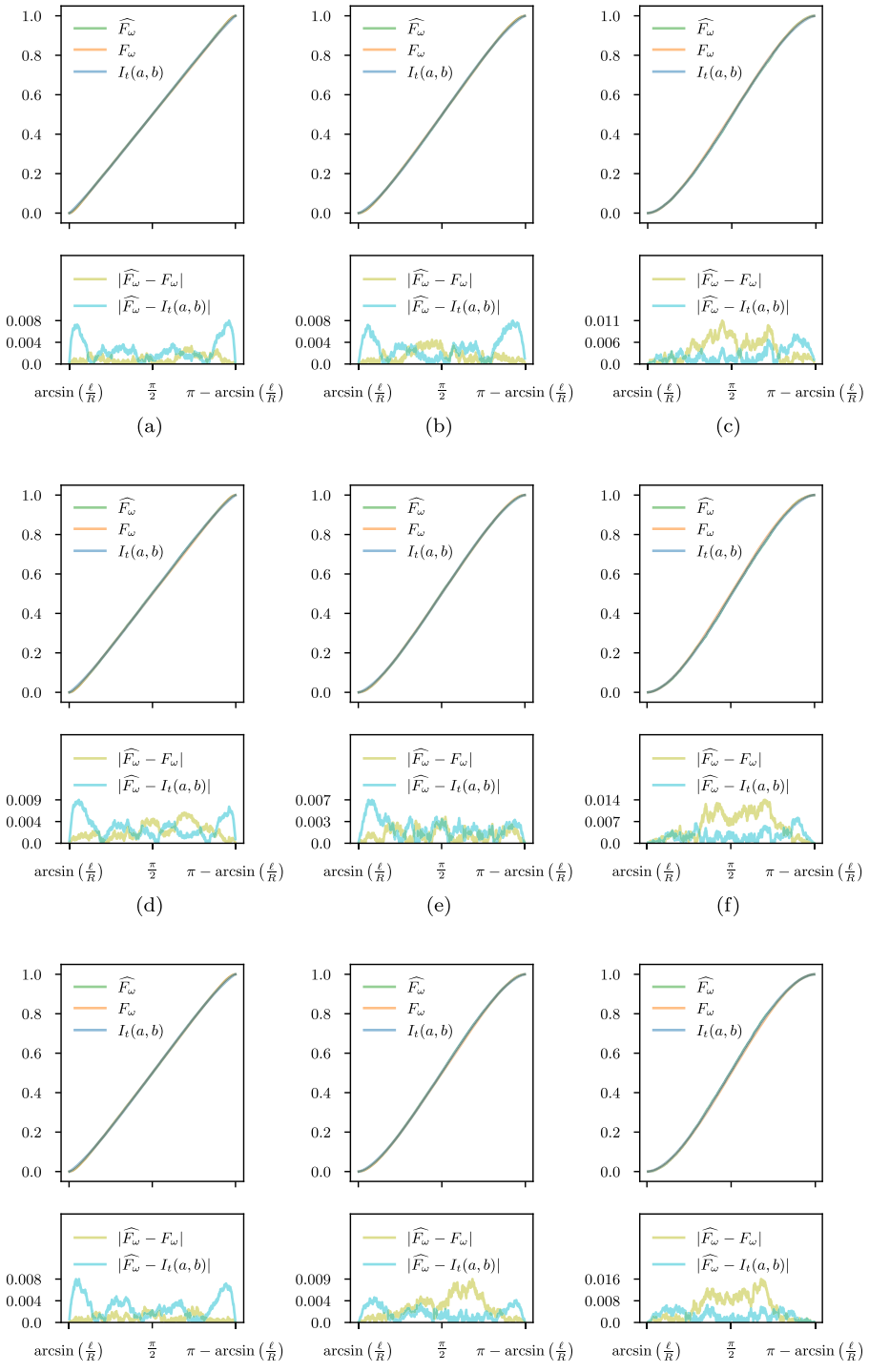
$$\begin{aligned} \rho^2 - \ell^2 &= \left(\frac{\ell}{\sin(t)}\right)^2 - \ell^2 \\ &= \ell^2\left(\frac{\sin^2(t) + \cos^2(t)}{\sin^2(t)} - \frac{\sin^2(t)}{\sin^2(t)}\right) \\ &= \frac{\ell^2}{\tan^2(t)}. \end{aligned}$$

We now turn to Int_2. $E_{t,\ell,\gamma,\rho} = \rho^2\sin^2(t) - \ell^2\cos^2(\gamma) \geq 0$ is true because $\ell/\sin(t) \leq \rho$ and $\cos(\gamma) \leq 1$, we get

$$\begin{aligned} \text{Int}_2 = -\frac{1}{2} \cdot \Bigg[&\rho^2 \arcsin\left(\frac{\cos(t)}{\sqrt{1 - \frac{\ell^2\cos^2(\gamma)}{\rho^2}}}\right) \\ &+ \frac{\ell^2\cos^2(\gamma)}{\tan(t)} \ln\left(\sin(t)\sqrt{E_{t,\ell,\gamma,\rho}} + \rho\sin^2(t)\right) \\ &- \ell^2\cos^2(\gamma) \arctan\left(\frac{\cos(t)\rho}{\sqrt{E_{t,\ell,\gamma,\rho}}}\right) \Bigg]_{\ell/\sin(t)}^{R}, \end{aligned}$$

and after evaluation, equation (56) is obtained.

For the case $\pi/2 \leq t \leq \pi - \arcsin(\ell/R)$, it is true, due to symmetry, that $\text{area}(S_t) = \text{area}(S) - \text{area}(S_{\pi - t})$ where area(S) can be computed in a similar way as area(S_t) for $\arcsin(\ell/R) \leq t < \pi/2$, but note that the integration is over the interval $[\arcsin(\ell/\rho), \pi - \arcsin(\ell/\rho)]$ with respect to ω, and $[\ell, R]$ with respect to ρ, respectively. Given the above, we leave it to the reader to verify equation (54).



Subfigures	(a, b)	$A(\widehat{F_\omega}, F_\omega)$	$A(\widehat{F_\omega}, I_t(a,b))$
(a), (b), (c)	(1.103, 1.105), (1.362, 1.357), (1.682, 1.656)	0.00093, 0.00134, 0.00382	0.0026, 0.00275, 0.00223
(d), (e), (f)	(1.195, 1.205), (1.484, 1.484), (1.776, 1.735)	0.00233, 0.0013, 0.00562	0.00305, 0.00229, 0.00232
(g), (h), (i)	(1.235, 1.235), (1.542, 1.561), (1.826, 1.867)	0.00083, 0.0029, 0.00526	0.00287, 0.0018, 0.00232

FIGURE 19. Differences between the functions F_ω, $\widehat{F_\omega}$ and the fitted function $I_t(a, b)$. The first row is for $\gamma = \pi/8$, the second row is for $\gamma = \pi/4$ and the third row is for $\gamma = 3\pi/8$. The first column is for $\ell/R = 0.2$, the second column is for $\ell/R = 0.5$ and third column is for $\ell/R = 0.8$. Below we report the parameters (a, b) obtained in the fitting and the (approximate) area rate $A(\widehat{F_\omega}, F_\omega)$ ($A(\widehat{F_\omega}, I_t(a, b))$) between the $\widehat{F_\omega}$ and F_ω ($I_t(a, b)$) functions.

For the case $\gamma = 0$, simply substituting 0 for γ in equations (54), (55) and (56) does not always work, since there is a division by zero. Therefore, in those cases, the values of the functions should be defined as limits where γ approaches zero from the right (because $\gamma \geq 0$). For example, to determine the formula for area(S), use

$$\lim_{\gamma \to 0^+} \ell^2 \sin^2(\gamma) \cdot \arctan\left(\frac{\sqrt{R^2 - \ell^2}}{\ell \sin(\gamma)}\right) = 0 \cdot (\pi/2) = 0.$$

The details are left to the reader. □

Theorem 3 gave a rather complicated formula for the function F_ω. Further simplifications could probably be made, but the main problem would still be that it has three parameters (R, ℓ, γ) and would be difficult to fit a function to it. Thus, using this formula, we would not be able to store the CDFs of the features in a concise form. However, the formula includes, among many other things, the function arcsin, which may remind us of the arcsine distribution and its CDF. This is a special case of the beta distribution, so the idea is to try to approximate the function F_ω by the CDF $I_t(a, b)$ of the beta distribution, where $I_t(a, b)$ denotes the regularized incomplete beta function with only parameters a, b.

By generating a point cloud that satisfies the case described at the beginning of this section (i.e., no noise and the plane is uniformly sampled), we can test how well the former idea works. Figure 19 illustrates the differences between the function F_ω obtained from the formula in Theorem 3, the empirical CDF $\widehat{F_\omega}$ obtained from the ω data, and the function obtained by fitting the regularized incomplete beta function $I_t(a, b)$ to the function $\widehat{F_\omega}$. In order to quantify the differences and to compare the results obtained with different choices γ and ℓ/R, we take the area under the absolute value of the differences between the functions, and divide it by $\pi - 2\arcsin(\ell/R)$ (i.e. the length of the interval where the values of the functions are between 0 and 1). We call this the area rate and denote it by A. (We will use the same notation in the following. Understandably, the interval will always vary, the length of which will be used to divide the area.) There are only small differences between the functions, as shown in this figure. It is also observed that for higher ℓ/R values the fitted function $I_t(a, b)$ is closer to the function $\widehat{F_\omega}$ than the function F_ω. This is probably because the functions sin, tan, arcsin, arctan etc. in the formula obtained by Theorem 3 are somewhat numerically imprecise and because there are many of them, these imprecisions are amplified. To summarize, we find that the function $I_t(a, b)$ approximates the function $\widehat{F_\omega}$ in a roughly similar way to the function obtained from Theorem 3. This result, in turn, encourages us to try to use the function $I_t(a, b)$ to approximate the CDFs of FPFH features.

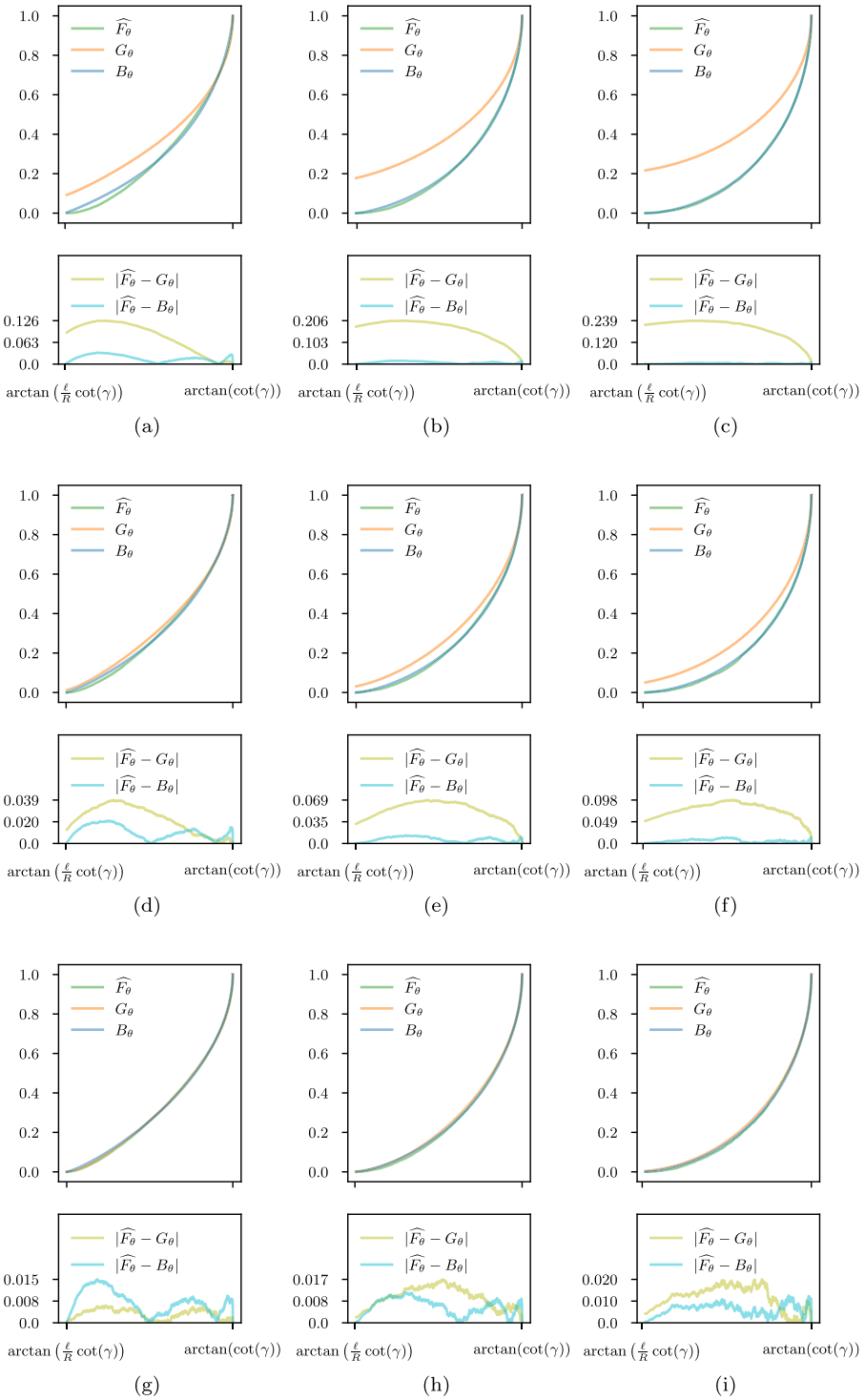
Importantly, the beta distribution is a notable family of probability distributions, not to be confused with function F_β we defined earlier, for which a formula could be derived in a similar way as we did for the function F_ω above. We do not consider this necessary because in fact the function F_β can also be approximated by the CDF of the beta distribution, which is not surprising based on Proposition 2. We will not show this separately, but will only show, using Proposition 1, how the functions F_θ and F_α can be approximated by the function $I_t(a, b)$. We also consider the function F_ϕ in the same way, using Theorem 2. Let us use the following notations (the definition of the function G_ϕ is given in equation (53)):

$$\begin{aligned}
G_\theta(t) &= 2 \cdot F^*_\beta(\arcsin(\tan(t)\tan(\gamma))) \\
B_\theta(t) &= I_{\frac{\arcsin(\tan(t)\tan(\gamma)) - \arcsin(\ell/R)}{\pi/2 - \arcsin(\ell/R)}}(a, b) \\
G_\alpha(t) &= F^*_\beta(\arccos(-t\cos(\gamma))) \\
B_\alpha(t) &= I_{\frac{\arccos(-t/\cos(\gamma)) - \arccos\left(\sqrt{1-\ell^2/R^2}\right)}{\arccos\left(-\sqrt{1-\ell^2/R^2}\right) - \arccos\left(\sqrt{1-\ell^2/R^2}\right)}}(a, b) \\
B_\phi(t) &= 1 - I_{\frac{\arcsin(t/\phi_0)}{\pi/2}}(a, b)
\end{aligned} \tag{60}$$

where the intervals for t are the same as in Proposition 1 and Theorem 2 (not copied here for brevity), parameters (a, b) are derived from curve fitting for specific cases. It is important to note that because of the domain of the function I, it is necessary to transform the values for the fit so that they fall within the interval $[0, 1]$. For this reason, subtractions and divisions are included in the formulas of functions $B_\bullet$ (for example, $\arcsin(\tan(t)\tan(\gamma))$ in B_θ becomes $\arcsin(\ell/R)$ for $t = \arctan(\cot(\gamma)\ell/R)$ and $\arcsin(1) = \pi/2$ for $t = \arctan(\cot(\gamma))$). The notation F^*_β used in the above formulas represents the average of the lower and upper estimates of F_β (obtained in Proposition 2):

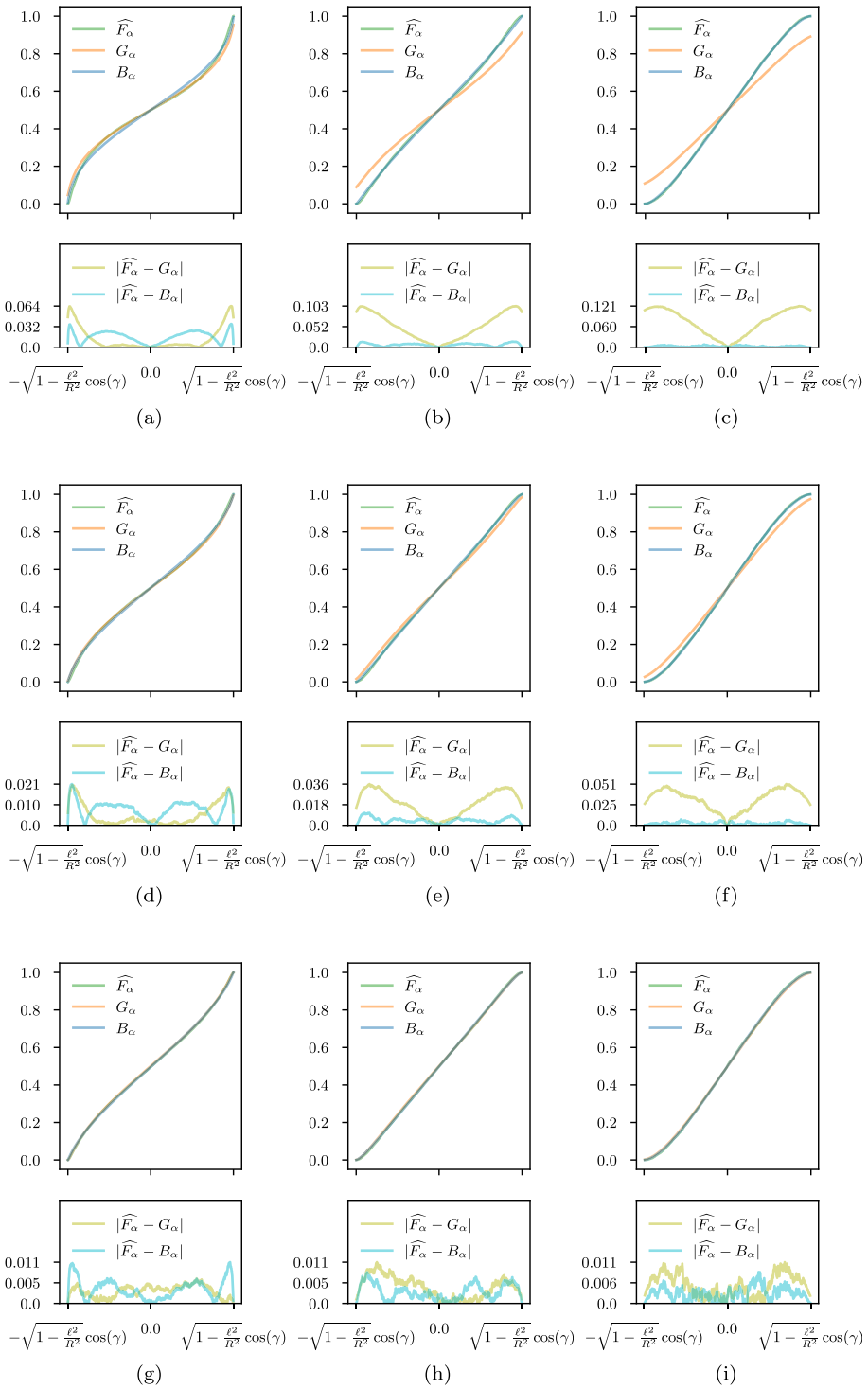
$$F^*_\beta(t) = \begin{cases} \frac{1}{2}(F_\omega(t) + F_\omega(\arctan(\tan(t)/\sin(\gamma)))) \\ \qquad \text{for } \arcsin(\ell/R) \leq t < \pi/2 \\ 1/2 \quad \text{for } t = \pi/2 \\ \frac{1}{2}(F_\omega(t) + F_\omega(\pi + \arctan(\tan(t)/\sin(\gamma)))) \\ \qquad \text{for } \pi/2 < t \leq \pi - \arcsin(\ell/R) \end{cases}$$

We use the same synthetic point clouds that we used for ω above, and examine how the functions $G_\bullet$ and $B_\bullet$ approximate the CDFs of the features. Based on Figure 20, we can conclude that the smaller the ratio ℓ/R and the larger the γ, the better the approximation of the function G_θ to the function $\widehat{F_\theta}$. (The more significant differences for smaller t are because the term of the upper estimate of function F_β for smaller t greatly "moves" the estimate.) In almost all cases, the function B_θ is a better approximation of the function $\widehat{F_\theta}$. Here it is true that function B_θ approximates more accurately for larger ratio ℓ/R and larger γ. From Figure 21, we can draw the conclusion that the situation is similar to θ, so that the smaller the ratio ℓ/R and the larger the γ, the better the approximation of the function G_α to the function $\widehat{F_\alpha}$. Again, the function B_α is a better approximation than the function G_α for larger ratio ℓ/R and larger γ. Figure 22 shows that function G_ϕ will always be an upper estimate of function $\widehat{F_\phi}$ and is spectacularly further away from it than function B_ϕ (hence the column $A(\widehat{F_\phi}, G_\phi)$ is missing from the table).



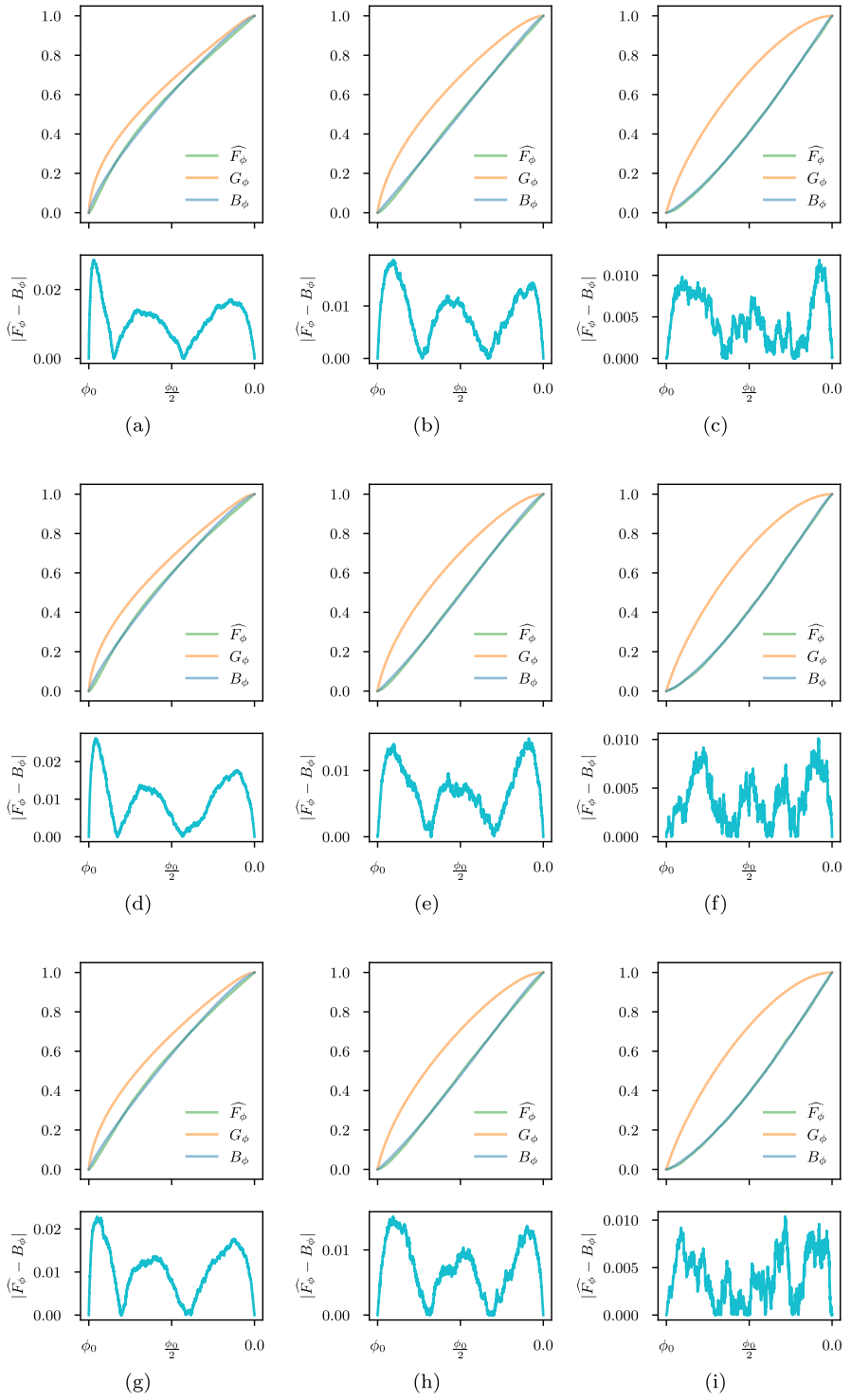
Subfigures	(a, b)	$A(\widehat{F_\theta}, G_\theta)$	$A(\widehat{F_\theta}, B_\theta)$
(a), (b), (c)	(1.045, 1.389), (1.288, 1.162), (1.565, 1.121)	0.07464, 0.16614, 0.20362	0.01666, 0.00786, 0.00353
(d), (e), (f)	(1.174, 1.188), (1.412, 1.126), (1.638, 1.078)	0.02177, 0.05228, 0.07501	0.01, 0.00631, 0.0056
(g), (h), (i)	(1.229, 1.099), (1.489, 1.105), (1.67, 1.104)	0.0031, 0.00976, 0.01155	0.00729, 0.00626, 0.00619

FIGURE 20. Differences between the functions $\widehat{F_\theta}$, G_θ and the fitted function B_θ. The first row is for $\gamma = \pi/8$, the second row is for $\gamma = \pi/4$ and the third row is for $\gamma = 3\pi/8$. The first column is for $\ell/R = 0.2$, the second column is for $\ell/R = 0.5$ and third column is for $\ell/R = 0.8$. Below we report the parameters (a, b) obtained in the fitting and the (approximate) area rate $A(\widehat{F_\theta}, G_\theta)$ ($A(\widehat{F_\theta}, B_\theta)$) between the $\widehat{F_\theta}$ and G_θ (B_θ) functions.



Subfigures	(a, b)	$A(\widehat{F_\alpha}, G_\alpha)$	$A(\widehat{F_\alpha}, B_\alpha)$
(a), (b), (c)	$(0.763, 0.762)$, $(1.163, 1.165)$, $(1.57, 1.575)$	0.01252, 0.05233, 0.07492	0.01618, 0.00649, 0.00297
(d), (e), (f)	$(1.011, 1.014)$, $(1.361, 1.361)$, $(1.77, 1.768)$	0.00535, 0.01954, 0.03178	0.00819, 0.00423, 0.00226
(g), (h), (i)	$(1.176, 1.163)$, $(1.502, 1.493)$, $(1.76, 1.76)$	0.00328, 0.0041, 0.00499	0.0038, 0.00296, 0.00272

FIGURE 21. Differences between the functions $\widehat{F_\alpha}$, G_α and the fitted function B_α. The first row is for $\gamma = \pi/8$, the second row is for $\gamma = \pi/4$ and the third row is for $\gamma = 3\pi/8$. The first column is for $\ell/R = 0.2$, the second column is for $\ell/R = 0.5$ and third column is for $\ell/R = 0.8$. Below we report the parameters (a, b) obtained in the fitting and the (approximate) area rate $A(\widehat{F_\alpha}, G_\alpha)$ $(A(\widehat{F_\alpha}, B_\alpha)$ between the $\widehat{F_\alpha}$ and G_α (B_α) functions.



Subfigures	(a, b)	$A(\widehat{F_\phi}, B_\phi)$
(a), (b), (c)	(1.21, 1.568), (1.186, 2.049), (1.117, 2.484)	0.01098, 0.00861, 0.00483
(d), (e), (f)	(1.281, 1.742), (1.252, 2.265), (1.179, 2.619)	0.01, 0.00719, 0.00388
(g), (h), (i)	(1.281, 1.792), (1.221, 2.274), (1.104, 2.57)	0.01065, 0.00724, 0.00393

FIGURE 22. Differences between the functions $\widehat{F_\phi}$, G_ϕ and the fitted function B_ϕ. The first row is for $\gamma = \pi/8$, the second row is for $\gamma = \pi/4$ and the third row is for $\gamma = 3\pi/8$. The first column is for $\ell/R = 0.2$, the second column is for $\ell/R = 0.5$ and third column is for $\ell/R = 0.8$. Below we report the parameters (a, b) obtained in the fitting and the (approximate) area rate $A(\widehat{F_\phi}, B_\phi)$ between the $\widehat{F_\phi}$ and B_ϕ functions.

Function B_ϕ, unlike before, approximates best for $\gamma = \pi/4$, and if γ decreases or increases in relation to this, it gives worse results. Furthermore, the larger the ratio ℓ/R, the better the estimate (similar to what was seen for the other two features). In summary, the functions $B_\bullet$ seem to approximate the CDFs of the features quite well.

APPENDIX C
HELLINGER DISTANCE COMPUTATION FOR FITTED CDFs OF FPFH FEATURES

In this section, the deduction of the Hellinger distance [62] for quantifying the similarity between two fitted CDFs of FPFH features is presented. Let us denote by χ one of the FPFH features (θ, α or ϕ). Let $H^2_\chi(p_1, p_2)$ be the squared Hellinger distance of the (estimated) probability distributions of the feature χ computed around points p_1 and p_2.

We will denote by $PA_{\chi,p}$ ($PB_{\chi,p}$) the array of parameters a (b) associated with the function $B_\chi^{[t_s,t_e]}$ fitted to the different intervals $[t_s, t_e]$ of the function $\widehat{F_\chi}$ computed around the point p (recall equations (14), (15), (16)). We write $PH_{\chi,p}$ for the array of $\widehat{F_\chi}(t_e) - \widehat{F_\chi}(t_s)$ values computed at different intervals $[t_s, t_e]$ which thus contains the "height" parameters h of the empirical function at different intervals. So the full fitted CDF B_χ for the point p will be the following using the notation:

$$B_{\chi,p}(t) = \begin{cases} 0 \quad \text{for } t < t_{s_0} \\ B_\chi^{[t_{s_0},t_{e_0}]}(t) \cdot PH_{\chi,p}[0] \quad \text{for } t \in [t_{s_0}, t_{e_0}] \\ B_\chi^{[t_{s_1},t_{e_1}]}(t) \cdot PH_{\chi,p}[1] + PH_{\chi,p}[0] \\ \quad \text{for } t \in [t_{s_1}, t_{e_1}] \\ \dots \\ B_\chi^{[t_{s_{L-1}},t_{e_{L-1}}]}(t) \cdot PH_{\chi,p}[L-1] + \sum_{k=0}^{L-2} PH_{\chi,p}[k] \\ \quad \text{for } t \in [t_{s_{L-1}}, t_{e_{L-1}}] \\ 1 \quad \text{for } t > t_{e_{L-1}} \end{cases} \tag{61}$$

where $[t_{s_0}, t_{e_0}]$, $[t_{s_1}, t_{e_1}]$, …, $[t_{s_{L-1}}, t_{e_{L-1}}]$ intervals are disjoint and their union yields the full domain. Let us denote by $B'_{\chi,p}(t)$ the derivative of the function $B_{\chi,p}(t)$. Denote by $f_k(t)$ one of the functions used in the argument of the regularized incomplete beta function associated with the different FPFH features (recall equation (13)) and fitted to the interval $[t_{s_k}, t_{e_k}]$ ($f_k(t_{s_k}) = 0$ and $f_k(t_{e_k}) = 1$, see equations (14), (15), (16)). We will write $f'_k(t)$ for the derivative of the function $f_k(t)$. It is easy to deduce for the point p that (62), as shown at the bottom of the page, where $B(\bullet, \bullet)$ denotes the beta function (using the usual notation).

Proposition 3: For the feature χ computed around points p_1 and p_2, it is true:

$$H^2_\chi(p_1, p_2) = 1 - \sum_{k=0}^{L-1} \sqrt{PC_{\chi,p_1,p_2}[k]} \cdot B\left(\overline{PA}_{\chi,p_1,p_2}[k], \overline{PB}_{\chi,p_1,p_2}[k]\right)$$

where

$$PC_{\chi,p_1,p_2}[k] = \frac{PH_{\chi,p_1}[k] \cdot PH_{\chi,p_2}[k]}{B\left(PA_{\chi,p_1}[k], PB_{\chi,p_1}[k]\right) \cdot B\left(PA_{\chi,p_2}[k], PB_{\chi,p_2}[k]\right)}$$

and

$$\overline{PA}_{\chi,p_1,p_2}[k] = \frac{PA_{\chi,p_1}[k] + PA_{\chi,p_2}[k]}{2},$$
$$\overline{PB}_{\chi,p_1,p_2}[k] = \frac{PB_{\chi,p_1}[k] + PB_{\chi,p_2}[k]}{2}.$$

Proof: By the definition of Hellinger distance and from equation (62), we conclude that

$$\begin{aligned} H^2_\chi(p_1, p_2) &= 1 - \int_{-\infty}^{\infty} \sqrt{B'_{\chi,p_1}(t) \cdot B'_{\chi,p_2}(t)}\, \mathrm{d}t \\ &= 1 - \sum_{k=0}^{L-1} \int_{[t_{s_k}, t_{e_k}]} \sqrt{PC_{\chi,p_1,p_2}[k]} \\ &\quad \cdot f_k(t)^{\overline{PA}_{\chi,p_1,p_2}[k]-1} \end{aligned}$$

$$B'_{\chi,p}(t) = \begin{cases} 0 \quad \text{for } t < t_{s_0} \\ \dfrac{PH_{\chi,p}[0] f_0(t)^{PA_{\chi,p}[0]-1}(1-f_0(t))^{PB_{\chi,p}[0]-1} f'_0(t)}{B\left(PA_{\chi,p}[0], PB_{\chi,p}[0]\right)} \\ \quad \text{for } t \in [t_{s_0}, t_{e_0}] \\ \dfrac{PH_{\chi,p}[1] f_1(t)^{PA_{\chi,p}[1]-1}(1-f_1(t))^{PB_{\chi,p}[1]-1} f'_1(t)}{B\left(PA_{\chi,p}[1], PB_{\chi,p}[1]\right)} \\ \quad \text{for } t \in [t_{s_1}, t_{e_1}] \\ \dots \\ \dfrac{PH_{\chi,p}[L-1] f_{L-1}(t)^{PA_{\chi,p}[L-1]-1}(1-f_{L-1}(t))^{PB_{\chi,p}[L-1]-1} f'_{L-1}(t)}{B\left(PA_{\chi,p}[L-1], PB_{\chi,p}[L-1]\right)} \\ \quad \text{for } t \in [t_{s_{L-1}}, t_{e_{L-1}}] \\ 0 \quad \text{for } t > t_{e_{L-1}} \end{cases} \tag{62}$$

$$
\begin{aligned}
&\cdot (1 - f_k(t))^{\overline{PB}_{\chi,p_1,p_2}[k]-1} \cdot f_k'(t)\,\mathrm{d}t \\
&= 1 - \sum_{k=0}^{L-1} \sqrt{PC_{\chi,p_1,p_2}[k]} \\
&\quad \cdot B\left(\overline{PA}_{\chi,p_1,p_2}[k], \overline{PB}_{\chi,p_1,p_2}[k]\right) \\
&\quad \cdot \Big(I_{f_k(t_{e_k})}\left(\overline{PA}_{\chi,p_1,p_2}[k], \overline{PB}_{\chi,p_1,p_2}[k]\right) \\
&\quad - I_{f_k(t_{s_k})}\left(\overline{PA}_{\chi,p_1,p_2}[k], \overline{PB}_{\chi,p_1,p_2}[k]\right)\Big) \\
&= 1 - \sum_{k=0}^{L-1} \sqrt{PC_{\chi,p_1,p_2}[k]} \\
&\quad \cdot B\left(\overline{PA}_{\chi,p_1,p_2}[k], \overline{PB}_{\chi,p_1,p_2}[k]\right) \\
&\quad \cdot \left(I_1\left(\overline{PA}_{\chi,p_1,p_2}[k], \overline{PB}_{\chi,p_1,p_2}[k]\right)\right. \\
&\quad \left. - I_0\left(\overline{PA}_{\chi,p_1,p_2}[k], \overline{PB}_{\chi,p_1,p_2}[k]\right)\right) \\
&= 1 - \sum_{k=0}^{L-1} \sqrt{PC_{\chi,p_1,p_2}[k]} \\
&\quad \cdot B\left(\overline{PA}_{\chi,p_1,p_2}[k], \overline{PB}_{\chi,p_1,p_2}[k]\right).
\end{aligned}
$$

□

The result of Proposition 3 is that the Hellinger distance between two fitted CDFs of feature χ for the points p_1 and p_2 can be computed directly from the parameter arrays PA_{χ,p_1}, PB_{χ,p_1}, PH_{χ,p_1}, PA_{χ,p_2}, PB_{χ,p_2}, PH_{χ,p_2}.

AUTHOR CONTRIBUTIONS

Conceptualization: János Márk Szalai-Gindl; methodology: János Márk Szalai-Gindl and Dániel Varga; formal analysis: János Márk Szalai-Gindl; investigation: János Márk Szalai-Gindl and Dániel Varga; writing—original draft preparation: János Márk Szalai-Gindl and Dániel Varga; writing—review and editing: János Márk Szalai-Gindl and Dániel Varga; visualization: János Márk Szalai-Gindl; project administration: János Márk Szalai-Gindl.

REFERENCES

[1] A. G. Buch and D. Kraft, "Local point pair feature histogram for accurate 3D matching," in *Proc. 29th Brit. Mach. Vis. Conf. (BMVC)*, Sep. 2018, pp. 1–12. [Online]. Available: http://bmvc2018.org/contents/papers/0576.pdf

[2] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3D object recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 998–1005.

[3] R. Vock, A. Dieckmann, S. Ochmann, and R. Klein, "Fast template matching and pose estimation in 3D point clouds," *Comput. Graph.*, vol. 79, pp. 36–45, Apr. 2019.

[4] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 3212–3217.

[5] J. Garstka and G. Peters, "Evaluation of local 3-D point cloud descriptors in terms of suitability for object classification," in *Proc. 13th Int. Conf. Informat. Control, Autom. Robot.*, 2016, pp. 540–547.

[6] F. Tombari, S. Salti, and L. D. Stefano, "Unique shape context for 3D data description," in *Proc. ACM Workshop 3D Object Retr.*, Oct. 2010, pp. 57–62.

[7] S. Quan, J. Ma, F. Hu, B. Fang, and T. Ma, "Local voxelized structure for 3D binary feature representation and robust registration of point clouds from low-cost sensors," *Inf. Sci.*, vol. 444, pp. 153–171, May 2018.

[8] R. Zhou, X. Li, and W. Jiang, "3D surface matching by a voxel-based buffer-weighted binary descriptor," *IEEE Access*, vol. 7, pp. 86635–86650, 2019.

[9] H. Deng, T. Birdal, and S. Ilic, "PPF-FoldNet: Unsupervised learning of rotation invariant 3D local descriptors," in *Proc. Europ. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2018, pp. 620–638.

[10] Z. Yew and G. Lee, "3DFeat-Net: Weakly supervised local 3D features for point cloud registration," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2018, pp. 630–646.

[11] M. Khoury, Q.-Y. Zhou, and V. Koltun, "Learning compact geometric features," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 153–161.

[12] Z. Gojcic, C. Zhou, J. D. Wegner, and A. Wieser, "The perfect match: 3D point cloud matching with smoothed densities," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5540–5549.

[13] C. Choy, J. Park, and V. Koltun, "Fully convolutional geometric features," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8957–8965.

[14] J. Du, R. Wang, and D. Cremers, "DH3D: Deep hierarchical 3D descriptors for robust large-scale 6DoF relocalization," in *Computer Vision–ECCV*. Cham, Switzerland: Springer, 2020, pp. 744–762.

[15] S. Ao, Q. Hu, B. Yang, A. Markham, and Y. Guo, "SpinNet: Learning a general surface descriptor for 3D point cloud registration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 11748–11757.

[16] J. Yang, S. Quan, P. Wang, and Y. Zhang, "Evaluating local geometric feature representations for 3D rigid data matching," *IEEE Trans. Image Process.*, vol. 29, pp. 2522–2535, 2020.

[17] X.-F. Han, Z.-A. Feng, S.-J. Sun, and G.-Q. Xiao, "3D point cloud descriptors: State-of-the-art," *Artif. Intell. Rev.*, vol. 56, no. 10, pp. 12033–12083, Apr. 2023.

[18] H. Kim and A. Hilton, "Evaluation of 3D feature descriptors for multi-modal data registration," in *Proc. Int. Conf. 3D Vis. - 3DV*, Jun. 2013, pp. 119–126.

[19] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, and N. M. Kwok, "A comprehensive performance evaluation of 3D local feature descriptors," *Int. J. Comput. Vis.*, vol. 116, no. 1, pp. 66–89, Apr. 2015.

[20] R. Hänsch, T. Weber, and O. Hellwich, "Comparison of 3D interest point detectors and descriptors for point cloud fusion," *ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci.*, vols. II–3, pp. 57–64, Aug. 2014.

[21] B. Jutzi and H. Gross, "Nearest neighbour classification on laser point clouds to gain object structures from buildings," in *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XXXVIII-1-4-7/W5. Göttingen, Germany: Copernicus Publications, 2009, pp. 207–213.

[22] E. Wahl, U. Hillenbrand, and G. Hirzinger, "Surflet-pair-relation histograms: A statistical 3D-shape representation for rapid classification," in *Proc. 4th Int. Conf. 3-D Digit. Imag. Model. 3DIM*, 2003, pp. 474–481.

[23] F. Stein and G. Medioni, "Structural indexing: Efficient 3-D object recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 125–145, Jun. 1992.

[24] C. S. Chua and R. Jarvis, "Point signatures: A new representation for 3D object recognition," *Int. J. Comput. Vis.*, vol. 25, no. 1, pp. 63–85, Oct. 1997.

[25] G. Hetzel, B. Leibe, P. Levi, and B. Schiele, "3D object recognition from range images using local feature histograms," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. CVPR*, vol. 3, 2001, pp. 394–399.

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

[27] D. Holz, A. E. Ichim, F. Tombari, R. B. Rusu, and S. Behnke, "Registration with the point cloud library: A modular framework for aligning in 3-D," *IEEE Robot. Autom. Mag.*, vol. 22, no. 4, pp. 110–124, Dec. 2015.

[28] A. Frome, D. Huber, R. Kolluri, T. Bülow, and J. Malik, "Recognizing objects in range data using regional point descriptors," in *Computer Vision—ECCV* (Lecture Notes in Computer Science). Berlin, Germany: Springer, 2004, pp. 224–237.

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

[30] Y. Guo, F. A. Sohel, M. Bennamoun, J. Wan, and M. Lu, "RoPS: A local feature descriptor for 3D rigid objects based on rotational projection statistics," in *Proc. 1st Int. Conf. Commun., Signal Process., Appl. (ICCSPA)*, Feb. 2013, pp. 1–6.
[31] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 1–4.
[32] R. Rusu, Z. Marton, N. Blodow, and M. Beetz, "Persistent point feature histograms for 3D point clouds," in *Proc. 10th Int. Conf. Intel Autonomous Syst (IAS-10)*, Baden-Baden, Germany, Jan. 2008, pp. 119–128.
[33] D. Varga, J. M. Szalai-Gindl, M. Ambrus-Dobai, and S. Laki, "QBB: Quantile-based binarization of 3D point cloud descriptors," *IEEE Access*, vol. 10, pp. 67839–67850, 2022.
[34] S. M. Prakhya, B. Liu, and W. Lin, "B-SHOT: A binary feature descriptor for fast and efficient keypoint matching on 3D point clouds," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 1929–1934.
[35] Y. Zhang, C. Li, B. Guo, C. Guo, and S. Zhang, "KDD: A kernel density based descriptor for 3D point clouds," *Pattern Recognit.*, vol. 111, Mar. 2021, Art. no. 107691.
[36] S. M. Prakhya, J. Lin, V. Chandrasekhar, W. Lin, and B. Liu, "3DHoPD: A fast low-dimensional 3-D descriptor," *IEEE Robot. Autom. Lett.*, vol. 2, no. 3, pp. 1472–1479, Jul. 2017.
[37] H. Zhao, M. Tang, and H. Ding, "HoPPF: A novel local surface descriptor for 3D object recognition," *Pattern Recognit.*, vol. 103, Jul. 2020, Art. no. 107272.
[38] S. Hinterstoisser, V. Lepetit, N. Rajkumar, and K. Konolige, "Going further with point pair features," in *Computer Vision—ECCV*. Cham, Switzerland: Springer, 2016, pp. 834–848.
[39] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, "3DMatch: Learning local geometric descriptors from RGB-D reconstructions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 199–208.
[40] M. D. Spivak, *A Comprehensive Introduction to Differential Geometry* (Curves on Surfaces), vol. 3, 3rd ed. Houston, TX, USA: Publish or Perish, 1999, ch. 4, pp. 187–232.
[41] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," 2018, *arXiv:1801.09847*.
[42] K. Klasing, D. Althoff, D. Wollherr, and M. Buss, "Comparison of surface normal estimation methods for range sensing applications," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 3206–3211.
[43] W. Wang, X. Lu, D. De Silva Edirimuni, X. Liu, and A. Robles-Kelly, "Deep point cloud normal estimation via triplet learning," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2022, pp. 1–6.
[44] J. Zhang, J.-J. Cao, H.-R. Zhu, D.-M. Yan, and X.-P. Liu, "Geometry guided deep surface normal estimation," *Comput.-Aided Design*, vol. 142, Jan. 2022, Art. no. 103119.
[45] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," in *Proc. 19th Annu. Conf. Comput. Graph. Interact. Techn.*, Jul. 1992, pp. 71–78.
[46] G. Metzer, R. Hanocka, D. Zorin, R. Giryes, D. Panozzo, and D. Cohen-Or, "Orienting point clouds with dipole propagation," *ACM Trans. Graph.*, vol. 40, no. 4, pp. 1–14, Jul. 2021.
[47] P. Guerrero, Y. Kleiman, M. Ovsjanikov, and N. J. Mitra, "PCPNet learning local shape properties from raw point clouds," *Comput. Graph. Forum*, vol. 37, no. 2, pp. 75–85, May 2018.
[48] R. Xu, Z. Dou, N. Wang, S. Xin, S. Chen, M. Jiang, X. Guo, W. Wang, and C. Tu, "Globally consistent normal orientation for point clouds by regularizing the winding-number field," *ACM Trans. Graph.*, vol. 42, no. 4, pp. 1–15, Jul. 2023.
[49] E. Wahl, "Rapid histogram-based scene interpretation in three-dimensional surface point clouds," Ph.D. thesis, Technische Universität München, Munich, Germany, 2007.
[50] G. Hua, M. Brown, and S. Winder, "Discriminant embedding for local image descriptors," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, 2007, pp. 1–8.
[51] Y. Guo, F. Sohel, M. Bennamoun, J. Wan, and M. Lu, "A novel local surface feature for 3D object recognition under clutter and occlusion," *Inf. Sci.*, vol. 293, pp. 196–213, Feb. 2015.
[52] S. M. Prakhya, B. Liu, W. Lin, K. Li, and Y. Xiao, "On creating low dimensional 3D feature descriptors with PCA," in *Proc. TENCON IEEE Region 10 Conf.*, Nov. 2017, pp. 315–320.
[53] D. Varga, J. M. Szalai-Gindl, and S. Laki, "The descriptiveness of feature descriptors with reduced dimensionality," in *Proc. Eur. Conf. Adv. Databases Inf. Syst.* Cham, Switzerland: Springer, 2021, pp. 317–322.
[54] S. Quan and J. Ma, "On shortened 3D local binary descriptors," *Inf. Sci.*, vol. 510, pp. 33–49, Feb. 2020.
[55] B. Glocker, S. Izadi, J. Shotton, and A. Criminisi, "Real-time RGB-D camera relocalization," in *Proc. IEEE Int. Symp. Mixed Augmented Reality (ISMAR)*, Oct. 2013, pp. 173–179.
[56] S. Choi, Q.-Y. Zhou, and V. Koltun, "Robust reconstruction of indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5556–5565.
[57] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.
[58] H. Deng, T. Birdal, and S. Ilic, "PPFNet: Global context aware local features for robust 3D point matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 195–205.
[59] M. J. D. Powell, "The Bobyqa algorithm for bound constrained optimization without derivatives," Univ. Cambridge, Cambridge, MA, USA, Cambridge NA Tech. Rep. NA2009/06, vol. 26, 2009, pp. 26–46.
[60] S. G. Johnson. (2024). *The Nlopt Nonlinear-optimization Package*. [Online]. Available: https://github.com/stevengj/nlopt
[61] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 77–85.
[62] E. Hellinger, "Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen," *J. für die reine und angewandte Mathematik*, vol. 136, pp. 210–271, Jul. 1909.

JÁNOS MÁRK SZALAI-GINDL received the M.Sc. degree in mathematics from Budapest University of Technology and Economics, Budapest, Hungary, and the Ph.D. degree from Eötvös Loránd University (ELTE), Budapest, in 2020. He wrote his dissertation on data-intensive methods for managing scientific data. He is currently an Assistant Professor with the Department of Information System, Faculty of Informatics, ELTE. His main research interests include scientific databases and data science. He has recently participated in multiple research projects in those fields.

DÁNIEL VARGA received the M.Sc. degree in computer science from Eötvös Loránd University (ELTE), Budapest, in 2017, and the Ph.D. degree, in 2023. He is currently an Assistant Professor with the Department of Information System, Faculty of Informatics, ELTE. His research interests include 3D point cloud processing related topics, such as surface reconstruction, point cloud matching, and object recognition.