

Persistent Point Feature Histograms for 3D Point Clouds

Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow and Michael Beetz

*Intelligent Autonomous Systems, Technische Universität München
Boltzmannstr. 3, Garching bei München, 85748*

Abstract.

This paper proposes a novel way of characterizing the local geometry of 3D points, using persistent feature histograms. The relationships between the neighbors of a point are analyzed and the resulted values are stored in a 16-bin histogram. The histograms are pose and point cloud density invariant and cope well with noisy datasets. We show that geometric primitives have unique signatures in this feature space, preserved even in the presence of additive noise. To extract a compact subset of points which characterizes a point cloud dataset, we perform an in-depth analysis of all point feature histograms using different distance metrics. Preliminary results show that point clouds can be roughly segmented based on the uniqueness of geometric primitives feature histograms. We validate our approach on datasets acquired from laser sensors in indoor (kitchen) environments.

Keywords. persistent feature histograms, point clouds, geometric reasoning

1. Introduction

Understanding a scene represented by point clouds can not be done directly and solely on the points' 3D coordinates. In particular, geometrical reasoning techniques can profit from compact, more informative point features, that represent the dataset better.

Estimated surface curvatures and normals for a point [1] are two of the most widely used point features, and play an important role in applications such as registration [2], or segmentation [3]. Both of them are considered *local features*, as they characterize a point using the information provided by the k closest neighbors of the point. Their values however, are highly sensitive to sensor noise and the selection of the k neighbors (i.e. if the k -neighborhood includes outliers, the estimation of the features will become erroneous). Robust feature descriptors such as moment invariants [4], spherical harmonic invariants [5], and integral volume descriptors [6] have been proposed as point features and used for registering partial scans of a model [7,6]. All of them are invariant to translation and 3D rotations, but are still sensitive to noise.

In general it is not clear how one should select the optimal k support for a point when computing any of the above mentioned features. If the data is highly contaminated with noise, selecting a small k will lead to large errors in the feature estimation. However, if k is too big, small details will be suppressed. Recently, work has been done on automatic computation of good k values (i.e. *scale*) for normal estimation on 3D point cloud

data [8,9] as well as principal curvatures [10,11,12] on multiple scales. Unfortunately, some of the above mentioned methods for computing an optimal scale require additional thresholds, such as d_1 and d_2 which are determined empirically in [8], and estimated using linear least-squares in [9] when knowledge about ground truth normal exists. In [10] the neighborhood is grown incrementally until a jump occurs in the variation-scale curve, but the method cannot be successfully applied to noisy point clouds, as the variations in the surface curvature are not modified smoothly with k . The selection of the T_c threshold in [11] is not intuitive, and the authors do not explain properly if the resulted persistent features are obtained using solely the intersection of features computed over different radii. The statistical estimation of curvatures in [12] uses a M-estimation framework to reject noise and outliers in the data and samples normal variations in an adaptively reweighted neighborhood, but it is unfortunately slow for large datasets, requiring approximately 20 minutes for about 10^6 points.

While the above mentioned descriptors can be considered good point features for some problems, they do not always represent enough information for characterizing a point, in the sense that they approximate a k -neighborhood of a point p with a single value. As a direct consequence, most scenes will contain many points with the same or very similar feature values, thus reducing their informative characteristics. Even if the feature estimation would be able to cope with noisy datasets, it can still be easily deducted that applications who rely on these 1D features will deal with multiple and false correspondences and will be prone to failure (e.g. registration). Alternatively, multiple-value point features such as curvature maps [13], or spin images [14], are some of the better local characterizations proposed for 3D meshes which got adopted for point cloud data. However, these representations require densely sampled data are not able to deal with the amount of noise usually present in 2^{1/2}D scans.

The 3D object recognition community has developed different methods for computing multi-value features which describe complete models for classification: curvature based histograms [15], spin image signatures [16], or surflet-pair-relation histograms [17]. All of them are based on the local estimation of surface normals and curvatures and describe the relationships between them by binning similar values into a global histogram. A high number of histograms per object is required by [15], but the method can cope with up to 20% occlusions. The 4D geometrical features used in [17] and the spin image signatures in [16] need a single histogram and achieve recognition rates over 90% with synthetic and CAD model datasets, and over 80% with added uniformly distributed noise levels below 1% [17]. All of the above show promising results, but since they have only been tested against synthetic range images, it's still unclear how they will perform when used on noisier real-world datasets.

We extend the work presented in [17] by computing local point feature histograms in 16D for each point in the cloud. We make an in-depth analysis of the points' signatures for different geometric primitives (i.e. planes, sphere, cylinders, etc), and reduce the theoretical computational complexity of the algorithm by a factor of approximately 2. The uniqueness of a feature point is analyzed by discretely sampling over an interval of sphere radii (for k -neighborhood selection). We statistically analyze different distance metrics between each point's histogram signature and the mean histogram of the cloud (μ -histogram), and select the points outside the $\mu \pm \alpha \cdot \sigma$ interval as persistent features.

Furthermore, we show that: a) our point feature histograms are: (i) robust in the presence of outliers and noisy data; (ii) pose and scale invariant; (iii) consistent over

different sampling densities in separate scans; and b) coupling them with persistence analysis yields accurate, and informative salient point features.

The remainder of the paper is organized as follows. Section 2 presents our implementation for computing point feature histograms, while Section 3 analyzes the histograms persistence over a spatial domain. We discuss experimental results in Section 4, and conclude with Section 5.

2. Point Feature Histograms

A problem that arises in point correspondence searches, is that the features usually used (e.g. surface curvature changes or integral volume descriptors) do not fully represent the underlying surface on which the point's neighborhood is positioned.

In order to efficiently obtain informative features, we propose the computation and usage of a histogram of values [18] which encodes the neighborhood's geometrical properties much better, and provides an overall point density and pose invariant multi-value feature. The histogram generalizes the mean surface curvature at a point p .

The input data consists of 3D $\{x, y, z\}$ point coordinates. For a given radius r , the algorithm will first estimate the surface normals at each point p by performing Principal Component Analysis (PCA) on the k -neighborhood defined by a sphere centered at p with radius r . Once the normals are obtained and consistently re-oriented (see [19] for a general algorithm for consistent normal orientation propagation), the histogram for p will be computed using the four geometric features as proposed in [17]. For every pair of points p_i and p_j ($i \neq j, j < i$) in the k -neighborhood of p , and their estimated normals n_i and n_j , we select a source p_s and target p_t point, the source being the one having the smaller angle between the associated normal and the line connecting the points:

if $\text{acos}(\langle n_i, p_j - p_i \rangle) \leq \text{acos}(\langle n_j, p_i - p_j \rangle)$ $p_s = p_i, p_t = p_j$ else $p_s = p_j, p_t = p_i$

and then define the Darboux frame with the origin in the source point as: $u = n_s$, $v = (p_t - p_s) \times u$, and $w = u \times v$.

The four features are categorized using a 16-bin histogram, where each bin at index idx contains the percentage of the point pairs in the k -neighborhood which have their features in the interval defined by idx :

$$\left. \begin{array}{l} f_1 = v \cdot n_t \\ f_2 = \|p_t - p_s\| \\ f_3 = u \cdot (p_t - p_s) / f_2 \\ f_4 = \text{atan}(w \cdot n_t, u \cdot n_t) \end{array} \right\} \Rightarrow idx = \sum_{i=1}^{i \leq 4} \text{step}(s_i, f_i) \cdot 2^{i-1}$$

where $\text{step}(s, f)$ is defined as 0 if $f < s$ and 1 otherwise. This means that by setting s_i to the center of the definition interval of f_i (i.e. 0 for features f_1, f_3, f_4 and r for f_2) the algorithm classifies each feature of a $\{p_i, p_j\}$ pair in p 's vicinity in two categories, and save the percentage of pairs which have the same category for all features.

The four features are a measure of the angles between the points' normals and the distance vector between them. Because f_1 and f_3 are dot products between normalized vectors, they are in fact the cosine of the angles between the 3D vectors, thus their value

is between ± 1 , and 0 if they are perpendicular. Similarly, f_4 is the arctangent of the angle that n_t forms with w if projected on the plane defined by $u = n_t$ and w , so its value is between $\pm\pi/2$, and 0 if they are parallel.

The number of histogram bins that can be formed using these four geometric features is div^4 , where div is the number of subdivisions of the features' value range. In our implementation, by dividing the feature values in two parts (f_i smaller or greater than s_i), we obtain a total of $2^4 = 16$ bins as the total number of combinations between the four features. Because the number of bins increases exponentially by the power of 4, using more than two subdivisions would result in a large number of extra dimensions for each point (e.g. $3^4 = 81$ D), which makes the computational problem intractable.

Figure 1 illustrates the differences using our proposed 16D feature set between query points located on various geometric surfaces. The surfaces were synthetically generated to have similar scales, densities, and noise levels as our input real-world datasets. For each of the mentioned surfaces, a point was selected such that it lies: (i) on a plane, (ii) on the middle of an edge of a cube (normals pointing both inside and outside), (iii) on the lateral surface of a cylinder at half the height, and (iv) on a sphere. The 16D feature histogram was generated using all its neighbors inside a sphere with radius $r = 2cm$. The results show that the different geometrical properties of each surface produce unique signatures in the feature histograms space.

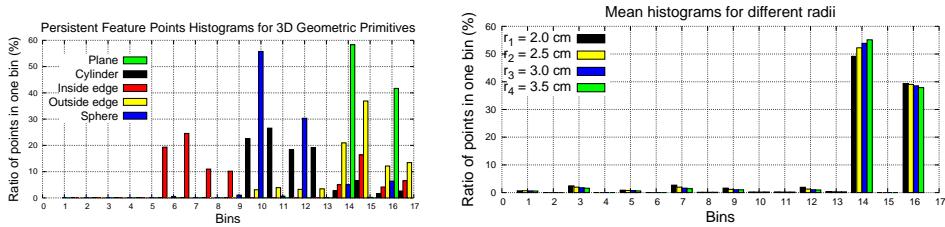


Figure 1. Feature Histograms for query points located on different geometric surfaces (left). Mean feature histograms over different radii for the kitchen scene (right).

Because of their properties, point feature histograms are promising to be more suitable candidates for problems like correspondence search while registering multiple scans under the same model. Figure 5 presents corresponding histogram features for similar points in two different overlapping point cloud datasets. In the k -neighborhood around point p , for each pair of points a source is uniquely defined. By implementing these restrictions (i.e. $i \neq j$ and $j < i$), the computational time for each point is reduced from the theoretical $c \cdot k^2$ to $c \cdot k \cdot (k-1)/2$, where c represents the time needed to compute the features for one pair.

3. Analyzing Feature Persistence

When characterizing a point cloud using point features, a compact subset of points P_f has to be found. The lesser the number of feature points and the better they approximate the data, the more efficient is the subsequent interpretation process. However, choosing the subset P_f is not easy, as it relies on a double dependency: both the number of neighbors k and the point cloud density φ . Our feature persistence analysis computes the subset

of points P_f , that minimizes the number of points considered for further analysis from the input data set. Corresponding points in different point cloud views of a scene will be likely to be found as persistent features in both scans, which helps in registration but also for segmenting similar points into regions.

In order to select the best feature points for a given cloud, we analyze the neighborhood of each point p multiple times, by enclosing the point on a sphere S with radius r_i and the point p as its center. We vary r over an interval depending on the point cloud size and density, and compute the local point feature histograms for every point. We then compute the mean of the feature histograms of all the points in the cloud (μ -histogram). By comparing the feature histogram of each point against the μ -histogram using a distance metric (see below), and building a distribution of distances, we can perform a statistical analysis of each feature's uniqueness over multiple radii. More specifically, we select the set of points (P_{f_i}) whose feature distances are outside the interval $\mu \pm \alpha \cdot \sigma$, as unique features. We do this for every r and at the end, select the persistent features which are unique in both r_i and r_{i+1} , that is:

$$P_f = \bigcup_{i=1}^{n-1} [P_{f_i} \cap P_{f_{i+1}}]$$

For comparing the point feature histograms with the μ -histogram of the cloud, we have performed an in-depth analysis using various distance metrics from literature (see Table 1), similar to [17,15]. The symbols p_i^f and μ_i represent the point feature histogram at bin i and the mean histogram of the entire dataset at bin i respectively.

Two of the most used distances in Euclidean spaces are the Manhattan (L1) and Euclidean (L2) norms, particularizations of the Minkowski p -distance:

$$\text{Manhattan (L1)} = \sum_{i=1}^{16} |p_i^f - \mu_i| \quad \text{Euclidean (L2)} = \sqrt{\sum_{i=1}^{16} (p_i^f - \mu_i)^2}$$

The Jeffries-Matusita (JM) metric (also known as Hellinger distance) is similar to the L2 (Euclidean) norm, but more sensitive to differences in smaller bins [20]:

$$\text{Jeffries-Matusita (JM)} = \sqrt{\sum_{i=1}^{16} (\sqrt{p_i^f} - \sqrt{\mu_i})^2}$$

The Bhattacharyya distance is widely used in statistics to measure the statistical separability of spectral classes:

$$\text{Bhattacharyya (B)} = -\ln \sum_{i=1}^{16} \sqrt{p_i^f \mu_i}$$

And finally two of the most popular measures for histogram matching in literature, the Chi-Square (χ^2) divergence and the Kullback-Leibler (KL) divergence:

$$\text{Chi-Square } (\chi^2) = \sum_{i=1}^{16} \frac{(p_i^f - \mu_i)^2}{p_i^f + \mu_i} \quad \text{KL divergence} = \sum_{i=1}^{16} (p_i^f - \mu_i) \cdot \ln \frac{p_i^f}{\mu_i}$$

The values of the r_i radii set are selected based on dimensionality of the features that need to be detected. Because small fine details are needed in our work at the expense of more data, (i.e. gaps between cupboard doors), we have selected $r_{min} = 2.0cm$ and $r_{max} = 3.5cm$. For our examples we fixed the value of α to 1, as only around 10 – 20% of the points are outside the $\mu \pm \sigma$ interval for different radii (see Figure 2 for an example),

thus selecting them as unique in the respective radius. By modifying the value of α one can roughly influence the number of persistent feature points resulting from the intersection and reunion operations. Figure 1 (right) shows the mean μ -histograms of the dataset for each radius. Notice how the resulting histograms are similar, and also very similar to the histogram of a point on a plane, telling us that most points in the dataset lie on planar surfaces. Because the deviation from the mean is small, selecting a small value of α is enough for identifying interesting points in the scan, as shown in Figure 5.

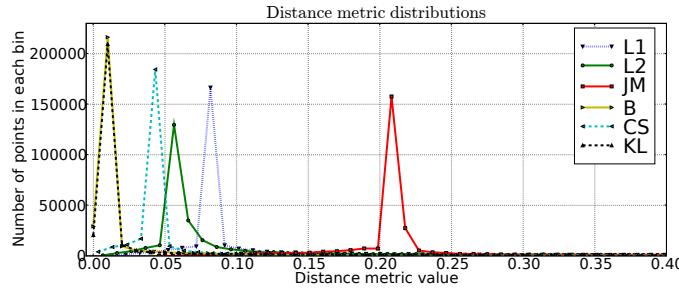


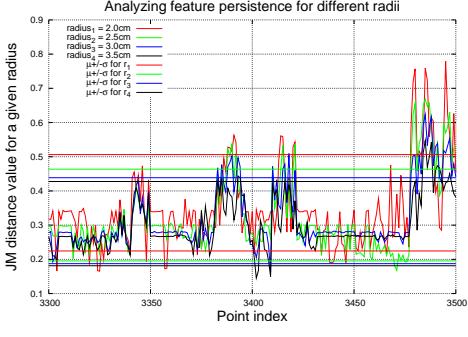
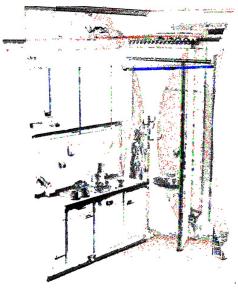
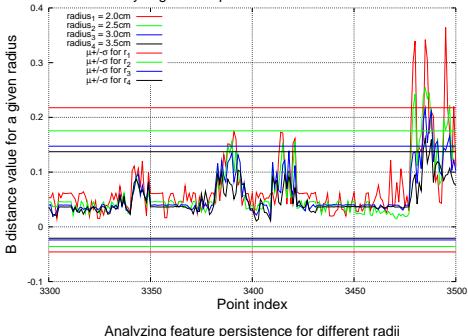
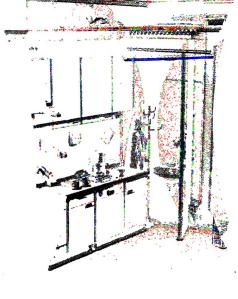
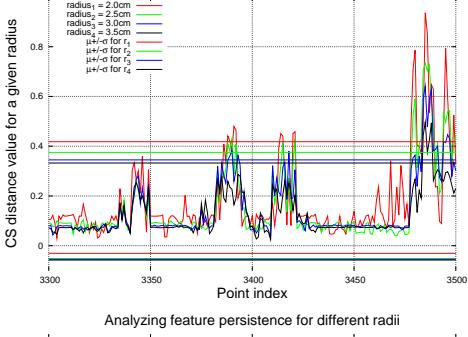
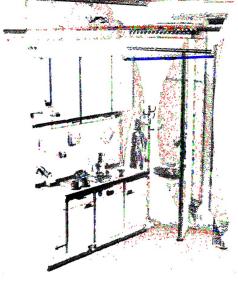
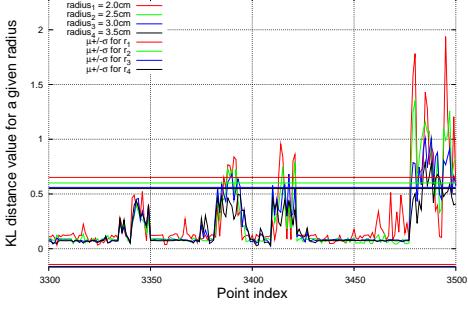
Figure 2. Distribution of feature histograms computed with different distance metrics for $r = 3\text{cm}$.

Table 1.: Analyzing the persistence of histogram features in a point cloud for 4 different radii (r_1 - red, r_2 - green, r_3 - blue, r_4 - black) on the left, and their appropriate distance graphs (for a narrow range of points for visualization purposes) on the right.

Metric	Unique features for each r	Distances from μ -histogram
Manhattan (L1)		
Euclidean (L2)		

Continued on next page

Table 1 – continued from previous page

Metric	Unique features for each r	Distances from μ -histogram
Jeffries-Matusita (JM)		
Bhattacharyya		
Chi-Square		
Kullback-Leibler		

4. Discussions and Experimental Results

Table 1 shows our persistence analysis on a point cloud using the presented distance metrics. We computed histograms for every point in the scan using 4 different sphere radii, and generated the mean histogram for each. For every radius, we used a different color to mark points in the scan whose distance to the respective mean histogram exceeded σ , and repeated this for the 6 distance metrics.

The s_1 , s_3 , and s_4 values are hard thresholds for angle values, thus they influence the decision of placing a value of $f_{1,3,4}$ in a certain bin. Since the computation of these values depends on the estimated surface normals, high degrees of noise can lead to small variations in the results. If the difference between two similar values is small, but they are on different sides of their respective threshold, the algorithm will place them in different bins. In our analysis pairs of points in a surface patch that lie on the same plane, or on perpendicular planes are particularly interesting. To ensure consistent histograms for planes, edges, and corners even under noisy conditions, we tolerate a deviation of $\pm 5^\circ$ in those features by selecting $s_1 = s_3 = s_4 = -5^\circ \approx -0.087$ radians. The resulted histograms become more robust in the presence of additive noise for the mentioned surfaces, without influencing significantly the other types of surface primitives.

To illustrate the above, we show the computed feature histograms for a point located in the middle of a planar patch of $10\text{cm} \times 10\text{cm}$, first on synthetically generated data without noise, and then on data with added zero-mean Gaussian noise with $\sigma = 0.1$ (see Figure 4). As shown in Figure 3, the estimated histograms are similar even under noisy conditions. Note that the histograms change only slightly as the noise level increases or the radius decreases, thus retaining enough similarity to the generic plane histogram.

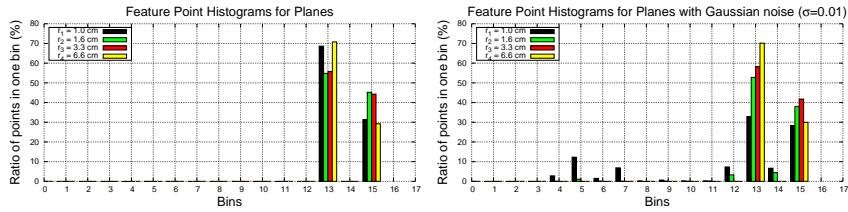


Figure 3. Feature Histograms over different radii for a point on a $10\text{cm} \times 10\text{cm}$ sized plane without noise – left; and with zero-mean Gaussian noise ($\sigma = 0.1$) – right.

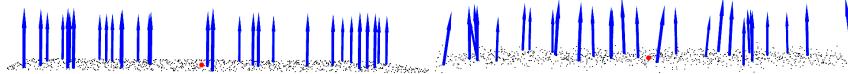


Figure 4. Point (red) for which histograms were computed on a plane without noise – left; and with noise – right. Normals showed for 1/3 of the points, computed for the smallest radius (1cm).

The validity of using feature histograms for registration is demonstrated in Figure 5. Points that are considered persistent features are marked in red. Note how the persistent feature analysis finds very similar points in the two scans. This speeds up applications like point cloud registration, since corresponding points are found easily and more robustly.

Another positive characteristic of the point feature histograms are their invariance to sampling density, due to the normalization of the histogram values with the number of point pairs in each k -neighborhood. A rough classification of points based on their

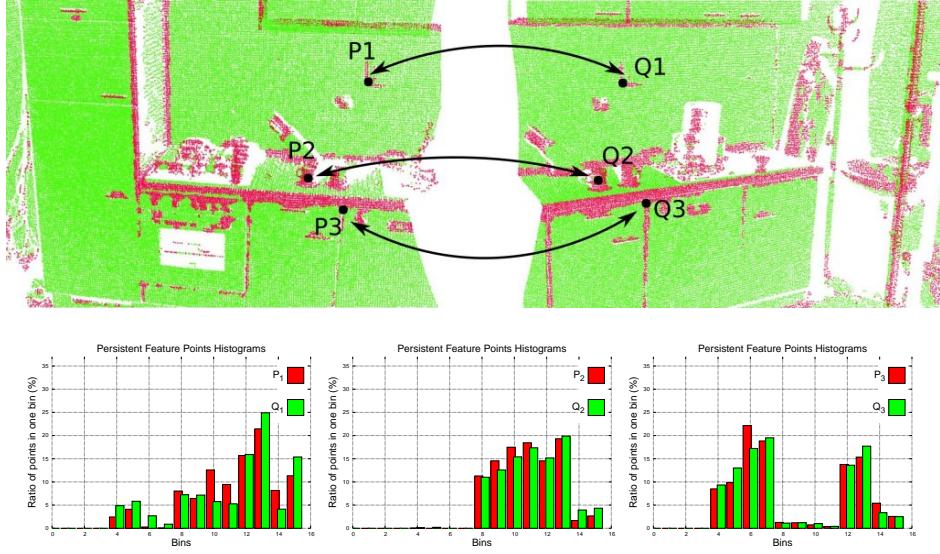


Figure 5. Feature Histograms for three pairs of corresponding points on different point cloud datasets.

histograms is shown in Figure 6, where the histograms of geometric primitive shapes (see Figure 1) are compared against all points. The test is performed using a simple distance metric, based on the comparison of the bin percentages at the location of the peaks in the geometric primitive shape histograms. This requires only few algebraic computations and comparisons for each point, i.e. $O(N)$.

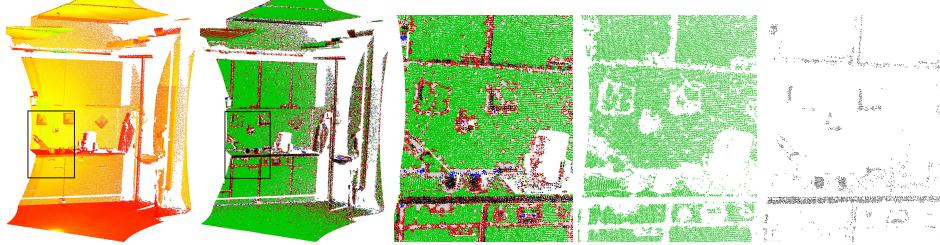


Figure 6. From left to right: partial scan of a kitchen; fast classification of each point's histogram (green: plane, red: edge, black: cylinder, blue: sphere, yellow: corner); detailed view of classification; points lying on planes and cylinders as determined by our method.

5. Conclusions

We have presented a method for computing feature histograms which characterize the local geometry at a given point p . The histograms are shown to be invariant to position, orientation, and point cloud density, and cope well with noisy datasets. The persistence of selected unique histograms is analyzed using different distance metrics over multiple scales, and a subset characterizing the input dataset is selected.

By creating unique signatures in this multi-dimensional space for points located on different surface types, the feature histograms show high potential for classifying and

segmenting point cloud surfaces. Experimental results show that the proposed approach looks promising for solving the problem of correspondence search in applications such as point cloud registration. While the registration and classification problems are not in this paper's scope, the presented simple examples are proofs of concept and show the discriminating power of the feature histograms.

References

- [1] A. M. and A. A., "On normals and projection operators for surfaces defined by point sets," in *Proceedings of Symposium on Point-Based Graphics 04*, 2004, pp. 149–155.
- [2] K.-H. Bae and D. D. Lichten, "Automated registration of unorganized point clouds from terrestrial laser scanners," in *International Archives of Photogrammetry and Remote Sensing (IAPRS)*, 2004, pp. 222–227.
- [3] T. Rabbani, F. van den Heuvel, and G. Vosselmann, "Segmentation of point clouds using smoothness constraint," in *IEVM06*, 2006.
- [4] F. Sadjadi and E. Hall, "Three-Dimensional Moment Invariants," *PAMI*, vol. 2, no. 2, pp. 127–136, 1980.
- [5] G. Burel and H. Hénocq, "Three-dimensional invariants and their application to object recognition," *Signal Process.*, vol. 45, no. 1, pp. 1–22, 1995.
- [6] N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann, "Robust Global Registration," in *Proc. Symp. Geom. Processing*, 2005.
- [7] G. Sharp, S. Lee, and D. Wehe, "ICP registration using invariant features," *IEEE Trans. on PAMI*, 24(1):90–102, 2002., 2002.
- [8] N. J. Mitra and A. Nguyen, "Estimating surface normals in noisy point cloud data," in *SCG '03: Proceedings of the nineteenth annual symposium on Computational geometry*, 2003, pp. 322–328.
- [9] J.-F. Lalonde, R. Unnikrishnan, N. Vandapel, and M. Hebert, "Scale Selection for Classification of Point-sampled 3-D Surfaces," in *Fifth International Conference on 3-D Digital Imaging and Modeling (3DIM 2005)*, June 2005, pp. 285 – 292.
- [10] M. Pauly, R. Keiser, and M. Gross, "Multi-scale feature extraction on point-sampled surfaces," pp. 281–289, 2003.
- [11] Y.-L. Yang, Y.-K. Lai, S.-M. Hu, and H. Pottmann, "Robust principal curvatures on multiple scales," in *SGP '06: Proceedings of the fourth Eurographics symposium on Geometry processing*, 2006, pp. 223–226.
- [12] E. Kalogerakis, P. Simari, D. Nowrouzezahrai, and K. Singh, "Robust statistical estimation of curvature on discretized surfaces," in *SGP '07: Proceedings of the fifth Eurographics symposium on Geometry processing*, 2007, pp. 13–22.
- [13] T. Gatzke, C. Grimm, M. Garland, and S. Zelinka, "Curvature Maps for Local Shape Comparison," in *SMI '05: Proceedings of the International Conference on Shape Modeling and Applications 2005 (SMI' 05)*, 2005, pp. 246–255.
- [14] A. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 433–449, May 1999.
- [15] G. Hetzel, B. Leibe, P. Levi, and B. Schiele, "3D Object Recognition from Range Images using Local Feature Histograms," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'01)*, vol. 2, 2001, pp. 394–399.
- [16] X. Li and I. Guskov, "3D object recognition from range images using pyramid matching," in *ICCV07*, 2007, pp. 1–6.
- [17] E. Wahl, U. Hillenbrand, and G. Hirzinger, "Surflet-Pair-Relation Histograms: A Statistical 3D-Shape Representation for Rapid Classification," in *3DIM03*, 2003, pp. 474–481.
- [18] R. B. Rusu, N. Blodow, Z. Marton, A. Soos, and M. Beetz, "Towards 3D Object Maps for Autonomous Household Robots," in *Proceedings of the 20th IEEE International Conference on Intelligent Robots and Systems (IROS), San Diego, CA, USA, Oct 29 - 2 Nov.*, 2007.
- [19] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," in *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, 1992, pp. 71–78.
- [20] L. Green and K.-M. Xu, "Comparison of Histograms for Use in Cloud Observation and Modeling," 2005.