Evaluation of Features for SVM-based Classification of Geometric Primitives in Point Clouds

Pascal Laube[1], Matthias O. Franz[1], Georg Umlauf[1]
[1]Institute for Optical Systems, University of Applied Sciences Constance, Germany

Abstract

In the reverse engineering process one has to classify parts of point clouds with the correct type of geometric primitive. Features based on different geometric properties like point relations, normals, and curvature information can be used to train classifiers like Support Vector Machines (SVM). These geometric features are estimated in the local neighborhood of a point of the point cloud. The multitude of different features makes an in-depth comparison necessary.
In this work we evaluate 23 features for the classification of geometric primitives in point clouds. Their performance is evaluated on SVMs when used to classify geometric primitives in simulated and real laser scanned point clouds. We also introduce a normalization of point cloud density to improve classification generalization.

1 Introduction

Reverse engineering (RE) of scanned 3d objects is the process of recovering a CAD representation approximating the acquired unstructured point data. RE consists of three main steps: pre-processing, segmentation, and fitting. Pre-processing includes e.g. subsampling or filtering the point cloud. The segmentation step yields patches in a point cloud that belong to the same parametric CAD model. Approaches like [1, 2, 3] use measures of similarity like smoothness or color to segment point clouds. These segmentation methods lack information about the underlying parametric model. For RE the resulting segments have to be classified as either geometric primitives (planes, spheres, cylinders, etc.) or free form surfaces. Depending on the classification result, in the fitting step a suitable CAD model is computed to approximate the point cloud patch.

In this paper we concentrate on the classification of geometric primitives prior to the fitting process. The considered geometric primitives are cones, planes, cylinders, ellipsoids, spheres, and tori.

A common approach in RE is to use a RANSAC based approach [4, 5]. Different parametric models are iteratively fitted and the one yielding the smallest error is assumed to be correct. RANSAC based algorithms have two problems. First, iterative fitting for each possible primitive type is expensive, if there is no initial knowledge of the type. Second, noisy point clouds might be approximated by the wrong primitive, because it yields the smallest error.

A different approach is to use local differential geometric properties and thresholds for classification [6]. This method requires user defined thresholds that may vary with different scanner types. Recently, machine learning approaches have superseded manual threshold definition. In [7] Support Vector Machines (SVM) together with curvature features are used to detect some types of geometric primitives. In [8] a set of feature histograms is used and their performance for k-nearest neighbor, k-means, and SVM classifiers are compared. In these papers only a small set of features is used. It is common to use all available features for classification, even if some of them might reduce classification performance.

In this paper we present an in-depth comparison of different feature descriptors. To our knowledge there exists no prior work on histogram arrangement, combination of features and evaluation of individual features especially in the context of geometry classification. We evaluate the performance of 19 individual features and their combinations. Our results give an indication on which features are meaningful for the RE process.

2 Geometric features

For the classification of the geometry of a point cloud $P = \{\mathbf{p}_i\}$ local geometric properties are used. These so-called geometric features are based on point coordinates, normal directions, and local curvature information. Most of the geometric features we use for our comparison, are well known in the fields of geometry processing.

For the purpose of using a SVM for the classification, the geometric features are computed for all $\mathbf{p}_i$ or a sufficiently large subset of P, see Section 2.1. These values are arranged in normalized feature histograms with varying numbers of bins. The concrete values for the normalization interval I and the number of bins b are shown in Table 1. For numerically sensitive geometric features the corresponding histograms are cropped to the $[0.05, 0.95]$ percentile to eliminate outliers.

The first four geometric features depend only on the location of the points in the point cloud and are adopted from [9]. For the computation of these geometric features random points from the point cloud are required. These points are mutually different, uniformly distributed random points from P.

F.1 *Point angles:* Angles between two vectors spanned by three random points.

F.2 *Point distances:* Euclidean distance δ between two random points.

F.3 *Centroid distances:* Euclidean distance of random points to the bounding box centroid.

F.4 *Triangle areas:* Square root of the triangle area of three random points.

F.5 *Cube cell count:* Number of points contained in 8 equally sized cells, which result from a uniform subdivision of the point cloud's bounding box. This feature is not invariant to rotation.

F.6 *K-Median points:* The coordinates of the medians resulting from clustering the point cloud into k clusters, to minimize the sum of distances of points in the cluster to the cluster median. Thus, the corresponding feature histogram is the concatenation of three coordinate-histograms. We used $k = 32$.

F.7 *Tetrahedron volumes:* Cubic root of the tetrahedron volume V of four random points $\mathbf{p}_1, \ldots, \mathbf{p}_4$

$$V = |(\mathbf{p}_1 - \mathbf{p}_4) \cdot ((\mathbf{p}_2 - \mathbf{p}_4) \times (\mathbf{p}_3 - \mathbf{p}_4))|/6.$$

Geometric features that do not only depend on point locations are normal angles and normal directions.

F.8 *Normal angles:* Angles between two normals at two random points.

F.9 *Normal directions:* Coordinates of the normalized normal at all points. Thus, the corresponding feature histogram is the concatenation of three coordinate-histograms.

To estimate the normal at point $\mathbf{p}$ in the point cloud, the set B of $\mathbf{p}$'s k-nearest neighbors is determined. Here, we used $k = 100$. The principal component analysis of B yields the covariance matrix, whose eigenvector $\mathbf{n}_\mathbf{p}$ corresponding to the smallest eigenvalue is used to estimate the normal at point $\mathbf{p}$.

The geometric features that depend on the curvature are defined as follows:

F.10 *Principal curvatures* κ_1, κ_2 are computed by polynomial fitting of osculating jets as in [10].

F.11 *Mean curvatures:* $H = \frac{1}{4}(\kappa_1 + \kappa_2)$.

F.12 *Gaussian curvatures:* $K = \kappa_1\kappa_2$.

F.13 *Curvature ratios:* $|\kappa_1/\kappa_2|$.

F.14 *Curvature changes:* Absolute difference between a random point's principal curvatures and those of its nearest neighbor and yield two concatenated histograms.

F.15 *Curvature angles:* Angles between the two corresponding principal curvature directions $\mathbf{v}_1$, $\mathbf{v}_2$ at two random points.

F.16 *Curvature directions:* Coordinates of two normalized principal curvature directions $\mathbf{v}_1$, $\mathbf{v}_2$ at all points. Thus, the corresponding feature histogram is the concatenation of six coordinate-histograms.

F.17 *Curvature differences:* Absolute differences of the principal curvatures, the Gaussian curvature, and the mean curvature at two random points, optionally weighted by distance.

F.18 *Shape index* as defined in [11] for $\kappa_1 > \kappa_2$

$$S_I = \frac{1}{2} - \frac{1}{\pi} \arctan \frac{\kappa_2 + \kappa_1}{\kappa_2 - \kappa_1}.$$

In order to combine the classification capabilities of individual geometric features, they can be combined into more general features. In [12] a combined normal based feature of two surflet pairs is proposed. These surflet pairs are defined as point-normal-pairs $(\mathbf{p}_1, \mathbf{n}_1)$ and $(\mathbf{p}_2, \mathbf{n}_2)$ with normalized normals $\mathbf{n}_1, \mathbf{n}_2$. From two surflet pairs a local, right-handed, orthonormal frame is computed

$$\mathbf{u} = \mathbf{n}_1, \quad \mathbf{v} = ((\mathbf{p}_2 - \mathbf{p}_1) \times \mathbf{u}) / \, |(\mathbf{p}_2 - \mathbf{p}_1) \times \mathbf{u}\|, \quad \mathbf{w} = \mathbf{u} \times \mathbf{v}.$$

This frame yields three geometric attributes

$$\alpha = \arctan(\mathbf{w} \cdot \mathbf{n}_2, \mathbf{u} \cdot \mathbf{n}_2), \quad \beta = \mathbf{v} \cdot \mathbf{n}_2, \quad \gamma = \mathbf{u} \cdot (\mathbf{p}_2 - \mathbf{p}_1)/\|\mathbf{p}_2 - \mathbf{p}_1\|.$$

Together with the point distance δ, these attributes define the surflet pair feature:

F.19 *Surflet pairs:* The tuple $(\alpha, \beta, \gamma, \delta)$ for two random points.

Further combined features can be constructed by concatenation of their respective histograms. Although any combination of the above features is possible, we combined only those features that proved most effective as individual features.

F.20 *Triple combination* of the best point-, normal-, and curvature-features: F.7, F.8, and F.15.

F.21 *Simple surflet combination* of F.7, F.8, and F.19.

F.22 *Extended surflet combination* of F.19 and F.20.

F.23 *All features combination* of features F.1,...,F.19.

2.1 Model selection

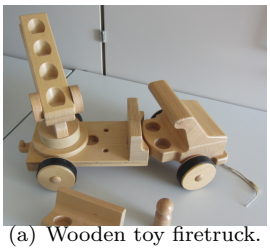
For the computation of training and test data we used the method of extracting patches of point clouds resulting from simulated scans of geometric primitives described in [13]. We also homogenize the density of the extracted patches. The density d_p of a patch $P = \{\mathbf{p}_1, \ldots, \mathbf{p}_n\}$ is computed as

$$d_P = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} \frac{\delta(\mathbf{p}_i, \mathbf{q}_j)}{k},$$

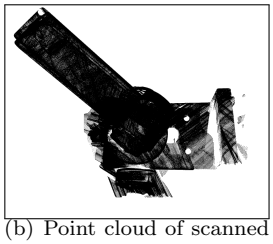
where $\mathbf{q}_j$ are the k-nearest neighbors of $\mathbf{p}_i$ and $\delta(\mathbf{p}_i, \mathbf{q}_j)$ is the Euclidean distance. Based on d_P and the target density d_t, a scaling factor $s = d_t/d_P$ is computed. Each point of P is scaled with s. We used $d_t = 0.01$.

The set of training data consists of $9,600$ point cloud patches of all six primitive classes with at least 150 points. 80% of these point clouds are used for training. The remaining 20% are used for feature evaluation by the true-positive-rate. To compute the geometric features from the point cloud patches often random point pairs, triplets, or quadruplets were chosen. In these cases 2^{17} feature values were sufficient to yield stable feature histograms.

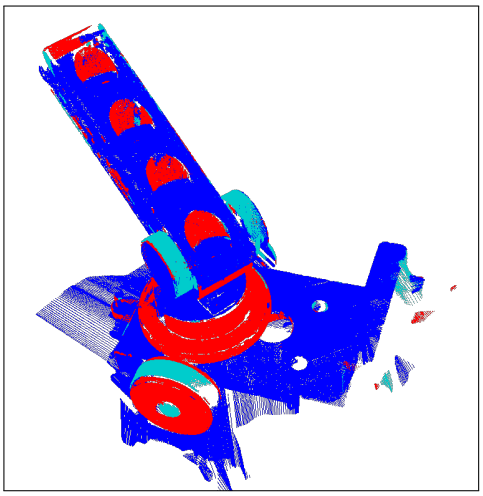
For the supervised learning we use SVMs. SVMs have shown to perform very well in high dimensional space when used with histograms [14]. For the SVM kernel we use a *Gaussian RBF Kernel*. A k-fold cross validation is used with $k = 5$. For optimizing the slack variable C and kernel size γ an extensive grid search is done. Given the d-class training data the one-versus-all approach uses d binary SVMs. For more details on SVM based learning methods refer to [15, 16].



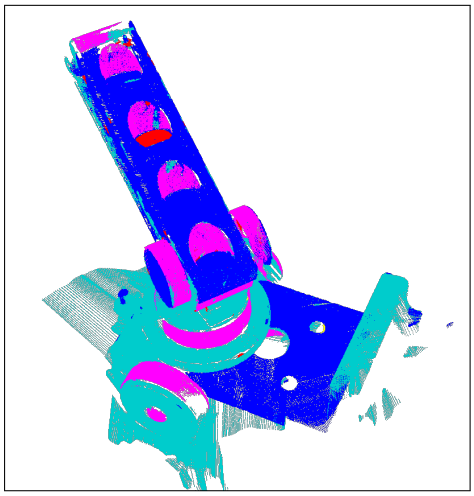
(a) Wooden toy firetruck.



(b) Point cloud of scanned firetruck.



(c) Point cloud of firetruck colored by primitive class.



(d) Point cloud of firetruck colored by primitive class without homogenization.

Figure 1: A real scan of a wooden toy firetruck colored by primitive class using the simple surflet combinations feature F.21: cones(red), planes (blue), cylinders (magenta), spheres (cyan), ellipsoids (green), tori (yellow).

3 Results

In this section we give the results of the feature evaluation process in terms of true-positive rates (TPR) of individual and combined features. All results are based on the same training data of point cloud patches from the six primitive classes. Table 1 shows the true-positive-rates for all features and corresponding values for C, γ, b, I, and a flag p, if the histogram is cropped to the $[0.05, 0.95]$ percentile. Note, that for F.23 the histograms are a combination of the respective feature histograms, which are normalized and cropped individually. Without the homogenization relative feature performance is identical to Table 1 but true-positive rates are about $10\% - 20\%$ higher.

Feature Histogram	TPR	C	γ	b	I	p
F.1 Point angles	0.564	50	2	64	$[0, 1]$	n
F.2 Point distances	0.437	10	5	64	$[0, 1]$	n
F.3 Centroid distances	0.370	100	0.1	64	$[0, 1]$	n
F.4 Triangle areas	0.515	10	10	64	$[0, 1]$	n
F.5 Cube cell count	0.420	5000	0.1	64	$[0, 1]$	n
F.6 K-Median points	0.240	1	0.1	96	$[0, 1]$	n
F.7 Tetrahedron vol.	0.588	1000	0.75	64	$[0, 1]$	n
F.8 Normal angles	0.529	10	1.5	64	$[0, 1]$	n
F.9 Normal directions	0.427	1	0.25	96	$[-1, 1]$	n
F.10 Principal curv.	0.308	0.1	1.5	128	$[0, 1]$	y
F.11 Mean curv.	0.280	5000	10	64	$[0, 1]$	n
F.12 Gaussian curv.	0.250	0.01	1	64	$[0, 1]$	n
F.13 Curv. ratio	0.286	0.1	2	64	$[0, 1]$	n
F.14 Curv. change	0.295	0.01	0.75	128	$[0, 1]$	n
F.15 Curv. angles	0.513	15	0.25	128	$[0, 1]$	n
F.16 Curv. directions	0.429	5	0.1	192	$[-1, 1]$	n
F.17 Curv. differences	0.351	0.1	5	128	$[0, 1]$	y
F.18 Shape index	0.280	0.1	2	64	$[0, 1]$	n
F.19 Surflet pairs	0.695	5000	0.1	128	$[0, 1]$	n
F.20 Triple combi.	0.632	1000	0.01	256	$[0, 1]$	n
F.21 Sim. surflet combi.	0.725	1000	0.1	256	$[0, 1]$	n
F.22 Ext. surflet combi.	0.711	5000	0.01	384	$[0, 1]$	n
F.23 All features combi.	0.369	10	0.01	1728	n/a	n/a

Table 1: Test results as true-positive rate (TPR) for all features of the simulated data set.

In contrast to a decrease in true-positive-rate, tests on real scans of point clouds show that homogenization leads to a better performance when classifying non-synthetic data. Figure 1(c) shows exemplar real scan data colored according to the detected primitives. Before coloring the point cloud has been segmented with a simple smoothness based region growing. We used an implementation of [17]. The scan has then been colored by using feature F.21 with (Figure 1(c)) and without (Figure 1(d)) homogenization.

4 Discussion

The results for the geometric features of Section 2 categorized as point-based, normal-based, curvature-based, and combined features are discussed separately. For observations on feature performance when classifying individual primitives we consulted feature confusion matrices, see Table 2.

The true-positive-rate of 0.588 of the tetrahedron volumes feature scores highest among the point-based features. While performing well for planes it confuses ellipsoids with spheres and tori with cylinders and has weak performance for cone classification. The reasons for this seems to be that point-based features do not capture curvature information sufficiently.

The normal angles feature F.8 has a classification rate of 0.529 and the highest classification rate among normal-based features. It is weak for classifying cones and cylinders. F.8 favors tori so that there is a large misclassification of cylinders as tori.

With 0.513 the curvature angle feature F.15 has the highest classification rate of the curvature-based features. It best classifies planes and has the weakest performance for ellipsoids, spheres and cylinders. Confusion of these primitive types can be observed for all features since they have strong geometrical relations especially when examined locally with noise.

The surflet pair feature F.19 performs best for planes, cylinders, and spheres, and has better performance for cones, ellipsoids, and tori than F.7, F.8, or F.15. The simple surflet combination feature F.21 performs best among all described features. It discriminates spheres, ellipsoids, cylinders, and tori best.

Based the true-positive-rates and color-coded real scans, we observe that homogenizing the density of the synthesized data decreases classification performance on the test set but improves classification applied on real scans. Since point cloud density is a result of the

	F.7 Tetrahedron volumes						F.8 Normal angles						F.15 Curvature angles						F.19 Surflet pairs						F.21 Simple surflet comb.					
	Cone	Plain	Cyl.	Ellips.	Sphere	Tori	Cone	Plain	Cyl.	Ellips.	Sphere	Tori	Cone	Plain	Cyl.	Ellips.	Sphere	Tori	Cone	Plain	Cyl.	Ellips.	Sphere	Tori	Cone	Plain	Cyl.	Ellips.	Sphere	Tori
Cone	0.19	0.4	0.09	0.15	0.3	0.07	0.3	0.36	0.08	0.17	0.21	0.06	0.32	0.31	0.1	0.16	0.17	0.11	0.58	0.22	0.1	0.06	0.13	0.02	0.67	0.15	0.11	0.06	0.09	0.02
Plain	0.02	1	0	0	0.07	0	0.03	1	0	0.05	0.01	0	0.05	1	0	0.02	0.01	0	0.03	1	0	0	0	0	0.02	1	0	0	0	0
Cyl.	0.02	0	0.47	0.18	0.01	0.47	0.09	0.01	0.35	0.2	0.06	0.42	0.40	0.02	0.41	0.18	0.03	0.45	0.05	0	0.74	0.13	0.01	0.16	0.04	0	0.77	0.09	0	0.17
Ellips.	0.02	0.06	0.09	0.61	0.26	0.11	0.08	0.1	0.09	0.53	0.17	0.17	0.09	0.14	0.16	0.41	0.14	0.2	0.05	0	0.06	0.7	0.16	0.11	0.03	0	0.06	0.74	0.15	0.09
Sphere	0.08	0.17	0	0.12	0.71	0	0.11	0.18	0.03	0.28	0.39	0.07	0.14	0.13	0.08	0.22	0.4	0.08	0.13	0.34	0	0.2	0.65	0	0.12	0.01	0	0.13	0.74	0
Tori	0.02	0	0.22	0.07	0	0.84	0.03	0	0.11	0.01	0.08	0.91	0.04	0.01	0.17	0.06	0.04	0.82	0.01	0	0.2	0.06	0	0.82	0.01	0	0.2	0.07	0	0.8

Table 2: Normalized confusion matrices for features F.7, F.8, F.15, F.19, and F.21 in heat-map coloring.

scanning process it affects features differently for different 3d-Scanners. Excluding density as an influence on feature computation by point cloud homogenization leads to better generalization of the trained classifier. Using the simple surflet combination feature F.21 we show an exemplar, colored real scan of a wooden toy firetruck, see Figure 1(c). The three main colors are red, blue, and cyan which correspond to cones, planes, and spheres, respectively. There are no tori or ellipsoids which is correct for the chosen object. Due to the dominance of cones in the selected feature, segments that might be cylindrical are classified as cones. Without homogenizing the density the classifier favors cylinders over cones which in some cases might be correct but often mistakes planes for spheres or cones, see Figure 1(d).

5 Conclusion

We present the evaluation of normal-, point-, and curvature-based features for primitive recognition in point clouds using support vector machines. Based on simulated scans we compare the performance of different features and feature combinations. Resulting classifiers were applied to real scans with and without homogenizing the density. Results of curvature-based features did not meet our expectations. Our results can be used to optimize the feature selection for the classification task at hand.

For future work we intend to use unsupervised learning methods, e.g. auto encoders, for feature engineering. To generate simulated scans that match real scans as close as possible is another aspect we plan to investigate.

Acknowledgments

This research has been funded by the Federal Ministry of Education and Research (BMBF) of Germany (project number 02P14A035)

References

[1] T. Rabbani, F. van den Heuvel, and G. Vosselmann, "Segmentation of point clouds using smoothness constraint," *Int. Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 36, no. 5, pp. 248–253, 2006.

[2] F. Moosmann, O. Pink, and C. Stiller, "Segmentation of 3d lidar data in non-flat urban environments using a local convexity criterion," in *Intelligent Vehicles Symposium*, 2009, pp. 215–220.

[3] Q. Zhana, Y. Liangb, and Y. Xiaoa, "Color-based segmentation of point clouds," *Int. Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 38, pp. 248–252, 2009.

[4] R. Schnabel, R. Wahl, and R. Klein, "Efficient RANSAC for point-cloud shape detection," *Computer graphics forum*, vol. 26, no. 2, pp. 214–226, 2007.

[5] C. Papazov and D. Burschka, "An efficient RANSAC for 3d object recognition in noisy and occluded scenes," in *Computer Vision ACCV 2010*, R. Kimmel, R. Klette, and A. Sugimoto, Eds., 2011, pp. 135–148.

[6] K. Denker, D. Hagel, J. Raible, G. Umlauf, and B. Hamann, "On-line reconstruction of CAD geometry," in *Intl. Conf. on 3D Vision*, 2013, pp. 151–158.

[7] G. Arbeiter, S. Fuchs, R. Bormann, J. Fischer, and A. Verl, "Evaluation of 3d feature descriptors for classification of surface geometries in point clouds," in *Int. Conf. on Intelligent Robots and Systems*, 2012, pp. 1644–1650.

[8] R. Rusu, Z. Marton, N. Blodow, and M. Beetz, "Learning informative point classes for the acquisition of object model maps," in *Int. Conf. on Control, Automation, Robotics and Vision*, 2008, pp. 643–650.

[9] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Shape distributions," *ACM Trans. on Graphics*, vol. 21, no. 4, pp. 807–832, 2002.

[10] F. Cazals and M. Pouget, "Estimating differential quantities using polynomial fitting of osculating jets," *Computer Aided Geometric Design*, vol. 22, no. 2, pp. 121–146, 2005.

[11] J. Koenderink and A. van Doorn, "Surface shape and curvature scales," *Image Vision Comput.*, vol. 10, no. 8, pp. 557–565, 1992.

[12] E. Wahl, U. Hillenbrand, and G. Hirzinger, "Surflet-pair-relation histograms: A statistical 3d-shape representation for rapid classification," in *3DIM*, 2003, pp. 474–482.

[13] M. Caputo, K. Denker, M. O. Franz, P. Laube, and G. Umlauf, "Support vector machines for classification of geometric primitives in point clouds," in *Int. Conf. on Curves and Surfaces*, 2014, pp. 80–95.

[14] O. Chapelle, P. Haffner, and V. Vapnik, "Support vector machines for histogram-based image classification," *IEEE Trans. on Neural Networks*, vol. 10, no. 5, pp. 1055–1064, 1999.

[15] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000.

[16] B. Schoelkopf and A. Smola, *Learning with Kernels*. MIT Press, 2001.

[17] R. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *Int. Conf. on Robotics and Automation*, 2011.