Learning geometric primitives in point clouds

Eurographics Symposium on Geometry Processing (2014)
Thomas Funkhouser and Shi-Min Hu (Guest Editors)

Learning Geometric Primitives in Point Clouds

M. Caputo[1], K. Denker[1], M. Franz[1], P. Laube[1], G. Umlauf[1]

[1]Institute for Optical Systems, University of Applied Sciences Constance, Germany

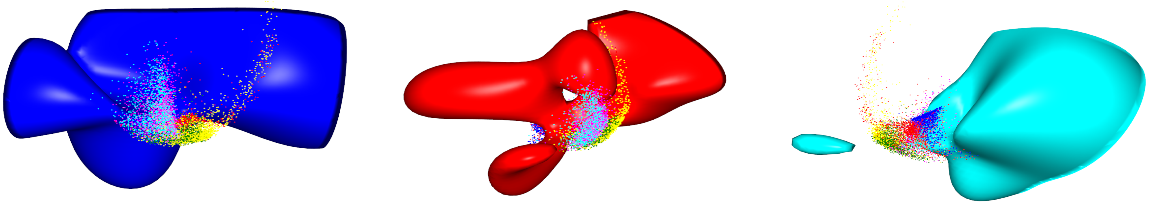

Figure 1: *Separating hyper-surfaces of the SVM to classify planes (blue), cones (red), and spheres (cyan) in the feature space of our feature F12. Remaining points refer to cylinders (green), ellipsoids (magenta), and tori (yellow).*

Abstract

Primitive recognition in 3D point clouds is an important aspect in reverse engineering. We propose a method for primitive recognition based on machine learning approaches. The machine learning approaches used for the classification are linear discriminant analysis (LDA) and multi-class support vector machines (SVM).

*For the classification process local geometric properties (*features*) of the point cloud are computed based on point relations, normals, and principal curvatures. For the training phase point clouds are generated using a simulation of a laser scanning device based on ray tracing with an error model. The classification rates of novel, curvature-based geometric features are compared to known geometric features to prove the effectiveness of the approach.*

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Pattern recognition]: Models—Geometric

1. Introduction

The reconstruction of CAD data from 3D scans has become more important in the last years. An important part of this reconstruction process is the detection of geometric primitives such as planes or cylinders within the scanned components. Our method works directly on the raw point cloud and does not require initial meshing. In [DHR*13] a system for online reconstruction is proposed, that uses a fixed set of heuristic rules for primitive detection based on estimation of local differential geometric properties. Since this system works only for planes, cylinders and spheres additional methods are required to detect other primitives such as cones or tori. Machine learning algorithms classify input samples and predict their class by learning to discriminate the different classes within certain feature spaces from training samples. These machine learning algorithms require fixed length feature vectors as input. We use pre-trained machine learning algorithms to label point cloud patches (Fig. 2) with their corresponding primitive class.

2. Machine learning methods and feature extraction

Machine learning algorithms such as linear discriminant analysis (LDA) or support-vector machines (SVM) need a training phase. In this phase the LDA and SVM are exposed to a large set of pre-classified point cloud patches to learn to discriminate between the different classes: plane, cylinder, cone, sphere, ellipsoid, and torus. The information about these classes is represented in distributions of geometric features exacted form the local geometry in the point cloud patch. Training point clouds are generated by a laser scanning simulation. This simulation is based on ray tracing to compute the intersections of a simulated laser line probe and a virtual geometric primitive. The laser scanner model in-

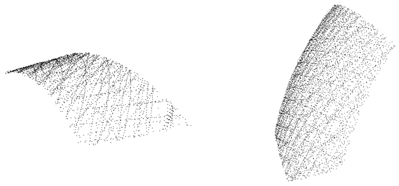

Figure 2: *Point cloud patches of a cylinder (left) and a sphere (right).*

cludes two types of systematic errors: A random offset added to the scanner position and to each scanned point. Both errors are normally distributed with zero mean and standard deviation $1/80$ and $1/400$, respectively. The extracted features are arranged in histograms with 2^i bins, $i = 6,\dots,9$. These histograms might be normalized and cropped to eliminate outliers. For SVM model selection is based on grid search. Fig. 1 shows some examples of hypersurfaces of trained SVMs. For details on SVM's see [CST00].

The four features F.1 *Point angles*, F.2 *Point distances*, F.3 *Triangle areas* and F.4 *Tetrahedron volumes* are based only on the location of the points in the point cloud, i.e. [OFCDk02]. The resulting histograms have 64 bins and are normalized to $[0,180]$ for F.1 and to $[0,1]$ for the others. Features that do not depend solely on point locations are F.5 *Normal angles* and F.6 *Normal directions*. For these normal-based features PCA is used to compute the normal of a point p based on it's 100-nearest neighbors. F.5 is computed as the angles between two normals at two random points. F.6 is given by the three coordinates of the normalized normal at all points. The histogram for F.6 is the concatenation of the three 32-bin histograms for the coordinates normalized to $[-1,1]$. Curvature based features are F.7 *Curvature angles*, F.8 *Curvature directions* and F.9 *Curvature differences*. For these features the principal curvatures and principal curvature directions were estimated by polynomial fitting of osculating jets [CP05]. F.7 is computed as the angles between the two corresponding principal curvature directions. F.8 is given by the six coordinates of two normalized principal curvature directions $\mathbf{v}_1$, $\mathbf{v}_2$ at all points. F.9 is computed from the absolute differences of the principal curvatures, the Gaussian, and mean curvature at two random points. F.7 results in two 64-bin histograms normalized to $[0,180]$ concatenated to one 128-bin histogram. F.8 is the concatenation of the six 32-bin histograms normalized to $[-1,1]$. The four 32-curvature-bin histograms of F.9 are cropped to the 0.05 to 0.95 percentile range, normalized, and concatenated. In [WHH03] a combined normal based feature is propose based on *surflet pairs* (F.10). It yields a histogram with 128 bins normalized to $[0,1]$. Additionally we combined those features that proved effective as individual features. *Triple combinations* (F.11) is combined from F.4, F.5 and F.7. *Simple surflet combinations* (F.12) is combined from F.4, F.5 and F.10. *Extended surflet combinations* (F.13) is combined from F.10 and F.11. The histograms for the combined features are concatenated from the histograms of the individual features. For comparison we tested nine additional features, e.g. the shape index [KvD92], which proved less effective in the experiments yielding a true-positive-rate below 0.5.

3. Results

The set of training data consists of 9600 point cloud patches of all six primitive classes with normally distributed parameters. 80% of these patches are used for training the remaining 20% to evaluate the method by the true-positive-rate. When using random point tuples for feature computation, 2^{17} tuples were sufficient to yield stable histograms. Table 1 shows the true-positive-rates for all features.

Feature Histogram	LDA	SVM
F.1 Point angles	0.643	0.713
F.2 Point distances	0.336	0.366
F.3 Triangle areas	0.594	0.644
F.4 Tetrahedron volumes	0.612	0.758
F.5 Normal angles	0.678	0.784
F.6 Normal directions	0.432	0.582
F.7 Curvature angles	0.628	0.695
F.8 Curvature directions	0.312	0.443
F.9 Curvature differences	0.356	0.420
F.10 Surflet pairs	0.720	0.850
F.11 Triple combinations	0.765	0.840
F.12 Simple surflet combinations	0.779	0.878
F.13 Extended surflet combinations	0.803	0.865

Table 1: *Test results as true positive rate for all learning algorithms and features of our artificial data set.*

4. Discussion and conclusion

In this poster we present a method for primitive recognition based on machine learning methods. It is based on point-, normal-, and curvature-based features for primitive classification. For future work we intend to extend the concept with segmentation, recognition of a lager set of geometries, and a comparison with algorithms like RANSAC.

References

[CP05] Cazals F., Pouget M.: Estimating differential quantities using polynomial fitting of osculating jets. *Computer Aided Geometric Design 22* (2005), 121–146. 2

[CST00] Cristianini N., Shawe-Taylor J.: *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000. 2

[DHR*13] Denker K., Hagel D., Raible J., Umlauf G., Hamann B.: On-line reconstruction of CAD geometry. In *Inter. Conf. on 3D Vision* (2013), pp. 151–158. 1

[KvD92] Koenderink J. J., van Doorn A. J.: Surface shape and curvature scales. *Image Vision Comp. 10* (1992), 557–565. 2

[OFCDk02] Osada R., Funkhouser T., Chazelle B., Dobkin D.: Shape distributions. *ACM TOG 21* (2002), 807–832. 2

[WHH03] Wahl E., Hillenbrand U., Hirzinger G.: Surflet-pair-relation histograms: A statistical 3d-shape representation for rapid classification. In *3DIM* (2003), IEEE, pp. 474–482. 2