Fast Geometric Point Labeling using Conditional Random Fields

Radu Bogdan Rusu, Andreas Holzbach, Nico Blodow, Michael Beetz
Intelligent Autonomous Systems, Computer Science Department, Technische Universität München
Boltzmannstr. 3, Garching bei München, 85748, Germany
{rusu,holzbach,blodow,beetz}@cs.tum.edu

***Abstract*— In this paper we present a new approach for labeling 3D points with different geometric surface primitives using a novel feature descriptor – the Fast Point Feature Histograms, and discriminative graphical models. To build informative and robust 3D feature point representations, our descriptors encode the underlying surface geometry around a point p using multi-value histograms. This highly dimensional feature space copes well with noisy sensor data and is not dependent on pose or sampling density. By defining classes of 3D geometric surfaces and making use of contextual information using Conditional Random Fields (CRFs), our system is able to successfully segment and label 3D point clouds, based on the type of surfaces the points are lying on. We validate and demonstrate the method's efficiency by comparing it against similar initiatives as well as present results for table setting datasets acquired in indoor environments.**

I. Introduction

Segmenting and interpreting the surrounding environment that a personal robot operates in from sensed 3D data is an important research topic. Besides recognizing a certain location on the map for localization or map refinement purposes, obtaining accurate and informative object models is essential for precise manipulation and grasping. Additionally, although the acquired data is discrete and represents only a few samples of the underlying scanned surfaces, it quickly becomes expensive to store and work with. It is therefore imperative to find ways to address this dimensionality problem and group clusters of points sampled from surfaces with similar geometrical properties together, or in some sense try to "classify the world". Achieving the latter annotates sensed 3D points and geometric structures with higher level semantics and will greatly simplify research in the aforementioned topics (e.g. manipulation and grasping).

In general, at the 3D point level, there are two basic alternatives for acquiring these annotations:

1) use a powerful, discriminative 3D feature descriptor, and learn different classes of surface or object types either by supervised or unsupervised learning, and then use the resultant model to classify newly acquired data;
2) use geometric reasoning techniques such as point cloud segmentation and region growing, in combination with robust estimators and non-linear optimization techniques to fit linear (e.g. planes, lines) and non-linear (e.g. cylinders, spheres) models to the data.

Both approaches have their advantages and disadvantages, but in most situations machine learning techniques – and thus the first approach, will outperform techniques purely based on geometric reasoning. The reason is that while linear models such as planes can be successfully found, fitting more complicated geometric primitives like cones becomes very difficult, due to noise, occlusions, and irregular density, but also due to the higher number of shape parameters that need to be found (increased complexity). To deal with such large solution spaces, heuristic hypotheses generators, while being unable to provide guarantees regarding the global optimum, can provide candidates which drastically reduce the number of models that need to be verified.

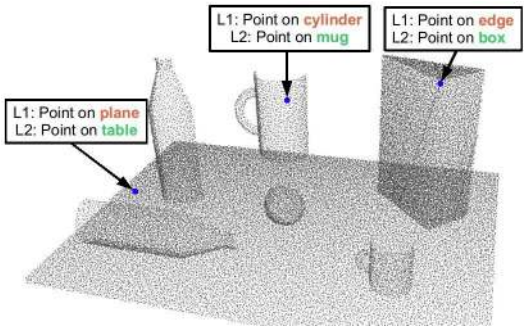


Fig. 1. An ideal 3D point based classification system providing two different point labels: geometry (L1) and object class (L2).

Additionally, we define a system able to provide two different hierarchical point annotation levels as an *ideal point classification system*. Because labeling objects lying on a table just with a class label, say *mug* versus *cereal box*, is not enough for manipulation and grasping (because different mugs have different shapes and sizes and need to be approached differently by the grasp planner), we require the classification system to annotate the local geometry around each point with classes of 3D geometric primitive shapes. This ensures that besides the object class, we are able to reconstruct the original object geometry and also plan better grasping points at the same time. Figure 1 presents the two required classification levels for 3 points sampled on separate objects with different local geometry.

This paper concentrates on the first annotation approach, and introduces a novel 3D feature descriptor for point clouds: the Fast Point Feature Histograms (FPFH). Our descriptors are based on the recently proposed Point Feature Histograms (PFH) [1] and robustly encode the local surface geometry around a point p using multi-value histograms, but reduce the original PFH computational complexity from $O(N^2)$ to

$O(N)$ while retaining most of their discriminative power. The FPFH descriptors cope well with noisy sensor data acquired from laser sensors and are independent of pose or sampling density. We present an in-depth analysis on their usage for accurate and reliable point cloud classification, which provides candidate region clusters for further parameterized geometric primitive fitting.

By defining classes of 3D geometric surfaces, such as: planes, edges, corners, cylinders, spheres, etc, and making use of contextual information using discriminative undirected graphical models (Conditional Random Fields), our system is able to successfully segment and label 3D points based on the type of surfaces the points are lying on. We view this is as an important building block which speeds up and improves the recognition and segmentation of objects in real world scenes.

As an application scenario, we will demonstrate the parameterization and usage of our system for the purpose of point classification for objects lying on tables in indoor environments. Figure 2 presents a classification snapshot for a scanned dataset representing a table with objects in a kitchen environment. The system outputs point labels for all objects found in the scene.

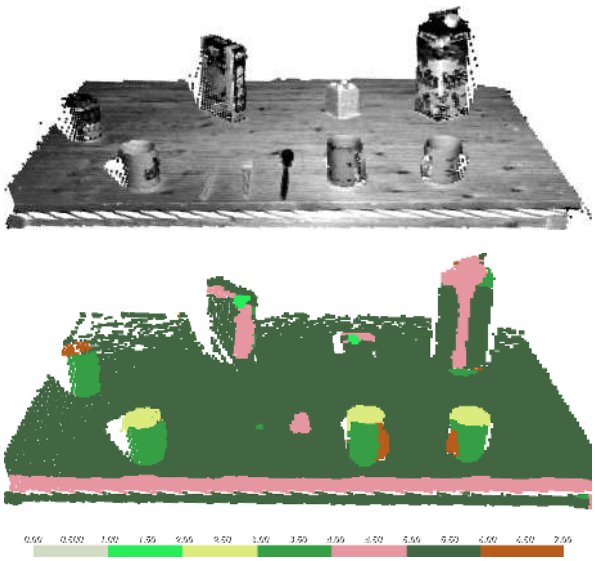


Fig. 2. Top: raw point cloud dataset acquired using the platform in Figure 7 containing approximatively 22000 points; bottom: classification results for points representing the table and the objects on it. The color legend is: dark green for planar, mid green for convex cylinders, yellow for concave cylinders, light green for corners, pink for convex edges and brown for convex tori. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The remainder of the paper is organized as follows. We address related work on similar initiatives in the next section, and describe the Fast Point Feature Histograms (FPFH) in Section III. The classes of 3D primitive geometric shapes used for learning the model are presented and analyzed in Section IV. We introduce our variant of Conditional Random Fields in Section V and discuss experimental results in Section VI. Finally we conclude and give insight into our future work in Section VII.

II. Related Work

The area of 3D point-based surface classification has seen significant progress over time, but more in recent years due to advances in electronics which fostered both better sensing devices as well as personal computers able to process the data faster. Suddenly algorithms for 3D feature estimation which were intractable for real world applications a few years ago became viable solutions for certain well-defined problems. Even so however, most of the work in 3D feature estimation has been done by the computer graphics community, on datasets usually acquired from high-end range scanners.

Two widely used geometric point features are the underlying surface's estimated curvature and normal [2], which have been also investigated for noisier point datasets [3]. Other proposals for point features include moment invariants and spherical harmonic invariants [4], and integral volume descriptors [5]. They have been successfully used in applications such as point cloud registration, and are known to be translation and rotation invariant, but they are not discriminative enough for determining the underlying surface type. Specialized point features that detect peaks, pits and saddle regions from Gaussian curvature thresholding for the problem of face detection and pose estimation are presented in [6]. Conformal factors are introduced by [7], as a measure of the total curvature of a shape at a given triangle vertex.

In general, descriptors that characterize points with a single value are not expressive enough to make the necessary distinctions for point-surface classification. As a direct consequence, most scenes will contain many points with the same or very similar feature values, thus reducing their informative characteristics. Alternatively, multiple-value point features such as curvature maps [8], or spin images [9], 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 and are not able to deal with the amount of noise usually present in 2.5D scans.

Following the work done by the computer vision community which developed reliable 2D feature descriptors, some preliminary work has been done recently in extending these descriptors to the 3D domain. In [10], a keypoint detector called THRIFT, based on 3D extensions of SURF [11] and SIFT [12], is used to detect repeated 3D structure in range data of building facades, however the authors do not address computational or scalability issues, and fail to compare their results with the state-of-the art. In [13], the authors propose RIFT, a rotation invariant 3D feature descriptor utilizing techniques from SIFT, and perform a comparison of their work with spin images, showing promising results for the problem of identifying corresponding points in 3D datasets.

We extend our previous work presented in [1] by computing fast local point feature histograms for each point in the cloud. We make an in-depth analysis of the points' signatures for different geometric primitives (i.e. planes, cylinders, toruses, etc) and show classification results using

two different machine learning methods: Support Vector Machines (SVM) and Conditional Random Fields (CRF).

The work in [14], [15] is similar to our labeling approach, as they too use probabilistic graphical models for the problem of point-cloud classification. In both, sets of simple point-based 3D features are defined, and a model is trained to classify points with respect to object classes such as: chairs, tables, screens, fans, and trash cans [15], respectively: wires, poles, ground, and scatter [14]. The main difference of our framework is that we do not craft sets of simpler 3D features for individual problems to solve, but propose the usage of a single well defined 3D descriptor for all.

III. Fast Point Feature Histograms (FPFH)

The computation of a Fast Point Feature Histogram (FPFH) at a point p relies on the point's 3D coordinates and the estimated surface normal at that point. As shown by [1], the PFH formulation is sufficient for capturing the local geometry around the point of interest, but if needed, is extensible to the use of other local properties such as color, curvature estimates, 2^{nd} order moment invariants, etc.

The main purpose of the FPFH formulation is to create a feature space in which 3D points lying on primitive geometric surfaces (e.g. planes, cylinders, spheres, etc) can be easily identified and labeled. As a requirement then, the discriminating power of the feature space has to be high enough so that points on the same surface type can be grouped in the same class, while points sampled on different surface types are assigned to different classes. In addition, the proposed feature space needs to be invariant to rigid transformations, and insensitive to point cloud density and noise to a certain degree.

To formulate the FPFH computational model, we introduce the following notations:

- $\boldsymbol{p}_i$ is a 3D point with $\{x_i, y_i, z_i\}$ geometric coordinates;
- $\boldsymbol{n}_i$ is a surface normal estimate at point p_i having a $\{nx_i, ny_i, nz_i\}$ direction;
- $\mathcal{P}_n = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots\}$ is a set of nD points (also represented by $\mathcal{P}$ for simplicity);
- $\mathcal{P}^k$ is the set of points $\boldsymbol{p}_j$, $(j \leq k)$, located in the k-neighborhood of a query point $\boldsymbol{p}_i$;
- $\|\cdot\|_x$ is the L_x norm (*e.g.* $\|\cdot\|_1$ is the Manhattan or L_1 norm, $\|\cdot\|_2$ is the Euclidean or L_2 norm).

Using the above formulation, the estimation of a FPFH descriptor includes the following steps: i) for each point $\boldsymbol{p}_i$, the set of $\mathcal{P}^k$ neighbors enclosed in the sphere with a given radius r are selected (k-neighborhood); ii) for every pair of points $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ $(i \neq j)$ in $\mathcal{P}^k$, and their estimated normals $\boldsymbol{n}_i$ and $\boldsymbol{n}_j$ ($\boldsymbol{p}_i$ being the point with a smaller angle between its associated normal and the line connecting the points), we define a Darboux uvn frame ($u = \boldsymbol{n}_i,\ v = (\boldsymbol{p}_j - \boldsymbol{p}_i) \times u,\ w = u \times v$) and compute the angular variations of $\boldsymbol{n}_i$ and $\boldsymbol{n}_j$ as follows:

$$\begin{aligned} \alpha &= v \cdot \boldsymbol{n}_j \\ \phi &= (u \cdot (\boldsymbol{p}_j - \boldsymbol{p}_i))/||\boldsymbol{p}_j - \boldsymbol{p}_i||_2 \\ \theta &= \arctan(w \cdot \boldsymbol{n}_j, u \cdot \boldsymbol{n}_j) \end{aligned} \quad (1)$$

Then, for all sets of computed $< \alpha, \phi, \theta >$ values in $\mathcal{P}^k$, a multi-value histogram is created as described in [1]. We call this the SPFH (Simplified Point Feature Histogram) as it only describes the direct relationships between a query point $\boldsymbol{p}_i$ and its k neighbors. To capture a more detailed representation of the underlying surface geometry at $\mathcal{P}^k$, in a second step we refine the final histogram (FPFH) of $\boldsymbol{p}_i$ by weighting all the neighboring SPFH values:

$$FPFH(\boldsymbol{p}_i) = SPFH(\boldsymbol{p}_i) + \frac{1}{k}\sum_{i=1}^{k}\frac{1}{\omega_k} \cdot SPFH(\boldsymbol{p}_k) \quad (2)$$

where the weight ω_k is chosen as the distance between the query point $\boldsymbol{p}_i$ and a neighboring point $\boldsymbol{p}_k$ in some metric space (Euclidean in our formulation), but could just as well be selected as a different measure if necessary. To understand the importance of this weighting scheme, Figure 3 presents the influence region diagram for a $\mathcal{P}^k$ set centered at $\boldsymbol{p}_q$. The SPFH connections are illustrated using red lines, linking $\boldsymbol{p}_i$ with its direct neighbors, while the extra FPFH connections (due to the additional weighting scheme) are shown in black.

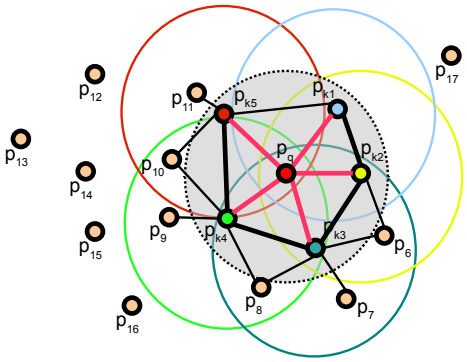


Fig. 3. The influence region diagram for a Fast Point Feature Histogram. Each query point (red) is connected only to its direct k-neighbors (enclosed by the gray circle). Each direct neighbor is connected to its own neighbors and the resulted histograms are weighted together with the histogram of the query point to form the FPFH. The thicker connections contribute to the FPFH twice.

The number of histogram bins that can be formed in a fully correlated FPFH space is given by q^d, where q is the number of *quantums* (i.e. subdivision intervals in a feature's value range) and d the number of features selected (e.g. in our case dividing each feature interval into 5 values will result in $5^3 = 125$ bins). In this space, a histogram bin increment corresponds to a point having certain values for all its 3 features. Because of this however, the resulting histogram will contain a lot of zero bins, and can thus contribute to a certain degree of information redundancy. A simplification of the above is to decorrelate the values, that is to simply create d separate histograms, one for each feature dimension in Equation 1, and then concatenate them together to create the FPFH.

In contrast to our PFH formulation [1], the FPFH descriptors reduce the computational complexity from $O(N^2)$

to $O(N)$. In the next section we analyze if the FPFH descriptors are informative enough to differentiate between points sampled on different 3D geometric shapes.

IV. 3D Geometric Primitives as Class Labels

A straightforward analysis of the FPFH discriminating power can be performed by looking at how features computed for different geometric surfaces resemble or differ from each other. Since the FPFH computation includes estimated surface normals, the features can be separately computed and grouped for both two cases of convex and concave shapes.

Figure 4 presents examples of FPFH signatures for points lying on 5 different convex surfaces, namely a cylinder, an edge, a corner, a torus, and finally a plane. To illustrate that the features are discriminative, in the left part of the figure we assembled a confusion matrix with gray values representing the distances between the mean histograms of the different shapes (a low distance is represented with white, while a high distance with black), obtained using the Histogram Intersection Kernel [16]:

$$d(FPFH_1, FPFH_2) = \sum_{i=1}^{N} \min(FPFH_1^i, FPFH_2^i) \quad (3)$$

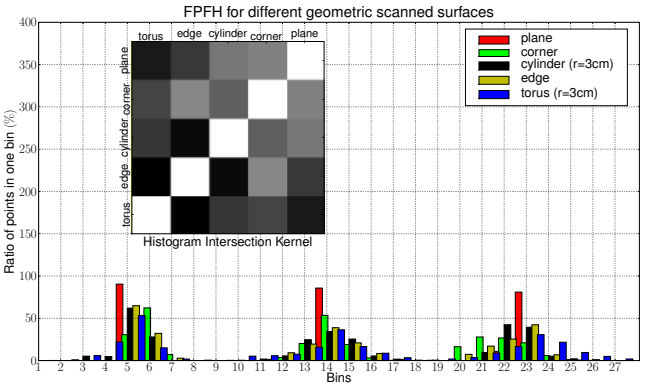


Fig. 4. Example of Fast Point Feature Histograms for points lying on primitive 3D geometric surfaces.

The results show that if the computational parameters are chosen carefully (i.e. the scale), the features are informative enough to differentiate between points lying on different geometric surfaces.

V. Conditional Random Fields

Generative graphical models, like Naive Bayes or Hidden Markov Models, represent a joint probability distribution $p(x, y)$, where x expresses the observations and y the classification label. Due to the inference problem this approach is not well applicable for fast labeling of multiple interacting features like in our case. A solution is to use discriminative models, such as Conditional Random Fields, which represent a conditional probability distribution $p(y|x)$. That means there is no need to model the observations and we can avoid making potentially erroneous independence assumptions among these features [17]. The probability $p(y|x)$ can be expressed as:

$$p(y|x) = \frac{p(y,x)}{p(x)} = \frac{p(y,x)}{\sum_{y'} p(y',x)} = \frac{\frac{1}{Z}\prod_{c\in C}\psi_c(x_c,y_c)}{\frac{1}{Z}\sum_{y'}\prod_{c\in C}\psi_c(x_c,y'_c)} \quad (4)$$

Therefore we can formulate a CRF model as:

$$p(y|x) = \frac{1}{Z(x)} \prod_{c\in C} \psi_c(x_c, y_c) \quad (5)$$

where $Z(x) = \sum_{y'} \prod_{c\in C} \psi_c(x_c, y')$. The factors ψ_c are potential functions of the random variables v_C within a clique $c \in C$ [18].

According to this mathematical definition, a CRF can be represented as a factor graph (see Figure 5). By defining the factors $\psi(y) = p(y)$ and $\psi(x, y) = p(x|y)$ we get an undirected graph with state and transition probabilities.

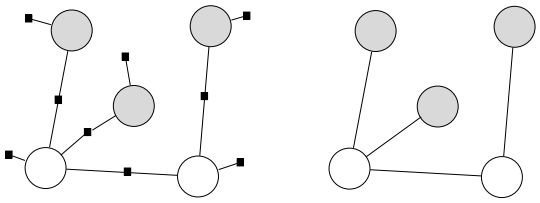


Fig. 5. Graphical representation of a CRF, as a factor graph (left), and as an undirected graph (right).

The potential functions ψ_c can be split into edge potentials ψ_{ij} and node potentials ψ_i as follows:

$$p(y|x) = \frac{1}{Z(x)} \prod_{(i,j)\in C} \psi_{ij}(y_i, y_j, x_i, x_j) \prod_{i=1}^{N} \psi_i(y_i, x_i) \quad (6)$$

where the node potentials are

$$\psi_i(y_i, x_i) = \exp\{\sum_L (\lambda_i^L x_i) y_i^L\} \quad (7)$$

and the edge potentials are

$$\psi_{ij}(y_i, y_j, x_i, x_j) = \exp\{\sum_L (\lambda_{ij} x_i x_j) y_i^L y_j^L\} \quad (8)$$

Learning in a Conditional Random Field is performed by estimating the node weights $\lambda_i = \{\lambda_i^1, \ldots, \lambda_i^L\}$ and the edge weights $\lambda_{ij} = \{\lambda_{ij}^1, \ldots, \lambda_{ij}^L\}$. The estimation is done by maximizing the log-likelihood of $p(y|x)$ [19]. One way for solving this nonlinear optimization problem is by applying the Broyden-Fletcher-Goldfarb-Shannon (BFGS) method. Specially Limited-memory BFGS gains fast results by approximating the inverse Hessian Matrix.

The given problem of labeling neighbored laser-scanned points motivates the exploitation of the graphical structure of CRFs. In our work, each conditioned node represents a laser-scanned point and each observation node a calculated feature. Figure 6 shows a simplified version of our model. The output nodes are connected with their observation nodes and the observation nodes of their surrounding neighbors, and each observation node is connected to its adjacent observation nodes.

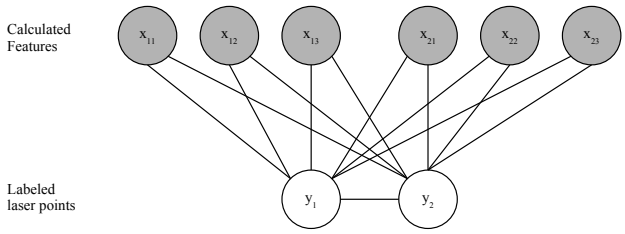


Fig. 6. Simplified version of our Conditional Random Field model

TABLE I

FEATURE ESTIMATION, MODEL LEARNING, AND TESTING RESULTS FOR THE DATASET IN FIGURE 2. THE METHOD INDICES REPRESENT THE TYPE OF FEATURES USED: PFH (^{1}) AND FPFH (^{2}). ALL COMPUTATIONS WERE PERFORMED ON A CORE2DUO @ 2 GHZ WITH 4 GB RAM.

Method	Feature Estimation (pts/s)	Model Training (s)	Model Testing (pts/s)	Accuracy
CRF^1	331.2	0.32	18926.17	89.58%
SVM^1	331.2	1.88	1807.11	90.13%
CRF^2	8173.4	0.091	76996.59	97.36%
SVM^2	8173.4	1.98	4704.90	89.67%

VI. DISCUSSIONS AND EXPERIMENTAL RESULTS

To validate our framework, we have performed several experiments of point cloud classification for real world datasets representing table scenes. The 3D point clouds were acquired using a SICK LMS400 laser scanner, mounted on a robot arm and sweeped in front of the table (see Figure 7). The average acquisition time was ≈1 s per complete 3D point cloud, and each dataset comprises around $70000 - 80000$ points. In total we acquired 10 such datasets with different objects and different arrangements (see Table II).

Fig. 7. Left: the robot used in our experiments; right: an example of one of the scanned table scenes.

Since we are only interested in classifying the points belonging to the objects on the table, we prefilter the raw point clouds by segmenting the table surface and retaining only the individual clusters lying on it. To do this, we assumed that the objects of interest are lying on the largest (biggest number of inliers) horizontal planar component in front of the robot, and we used a robust estimator to fit a plane model to it. Once the model was obtained, our method estimates the boundaries of the table and segments all object clusters supported by it (see Figure 8). Together with the table inliers, this constitutes the input to our feature estimation and surface classification experiments, and accounts for approximatively 22000-23000 points (≈ 30% of the original point cloud data).

Fig. 8. Left: raw point cloud dataset; right: the segmentation of the table (green) and the objects supported by it (random colors) from the rest of the dataset (black).

To learn a good model, we used object clusters from 3 of the data sets, and then tested the results on the rest. This is in contrast to our previous formulation in [1] where we trained the classifiers with synthetically noisified data. The training set was built by choosing primitive geometric surfaces out of the data sets, like corners, cylinders, edges, planes, or tori, both concave and convex. Our initial estimate is that these primitives account for over 95% data in scenes similar to ours.

We used the training data to compute two different models, using Support Vector Machines and Conditional Random Fields. Since ground truth was unavailable at the data acquisition stage, we manually labeled two scenes to test the individual accuracy of the two machine learning models trained. Additionally, on the same datasets, we estimated and tested two extra SVM and CRF models based on PFH features. Table I presents the computation time spent for both the feature estimation, and model training and testing parts.

As shown, the FPFH algorithm easily outperforms the PFH algorithm in computation performance. It calculates point histograms over 20 times faster while still gaining an edge in CRF classification accuracy. The training error curves for the CRF models for both PFH and FPFH are shown in Figure 9. Since the error norm becomes very small after approximatively 50 iterations, we stopped the model training there. Table II presents visual examples of some of the classified table scenes with both the SVM and CRF models.

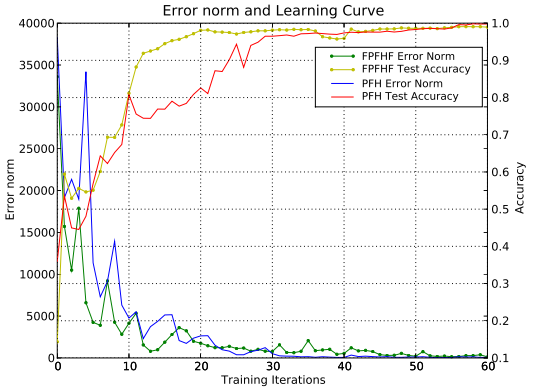
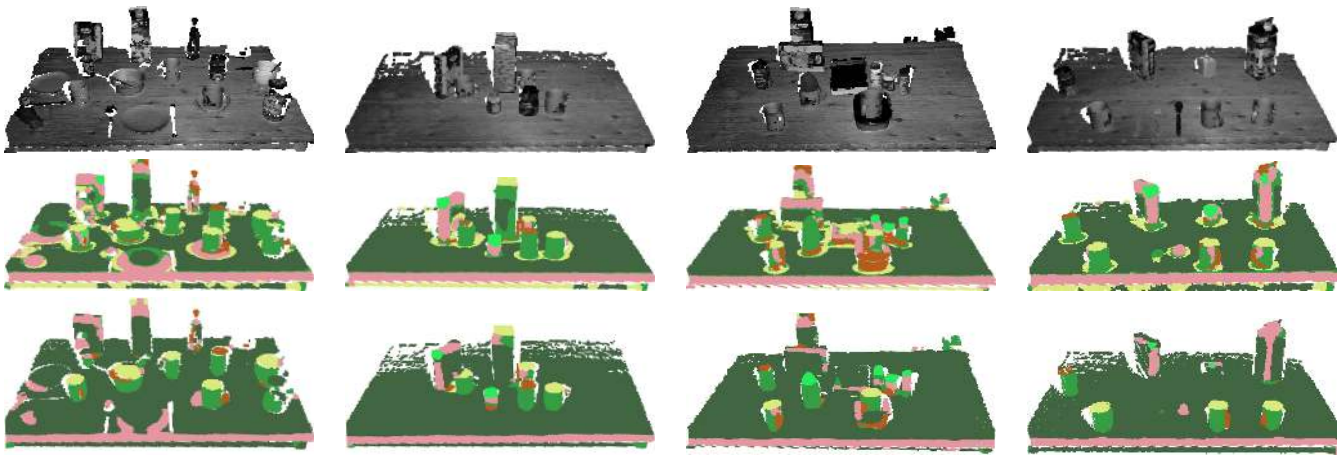


Fig. 9. Test accuracy and training error curves for the two CRF models.

TABLE II

A SUBSET OF 4 DIFFERENT DATASETS USED TO TEST THE LEARNED MODELS. THE IMAGES (TOP TO BOTTOM) REPRESENT: GRAYSCALE INTENSITY VALUES, CLASSIFICATION RESULTS OBTAINED WITH THE SVM MODEL, AND CLASSIFICATION RESULTS OBTAINED WITH THE CRF MODEL.

Scene 1	Scene 2	Scene 3	Scene 4



VII. CONCLUSIONS AND FUTURE WORK

In this paper we presented FPFH (Fast Point Feature Histogram), a novel and discriminative 3D feature descriptor, useful for the problem of point classification with respect to the underlying geometric surface primitive the point is lying on. By carefully defining 3D shape primitives, and making use of contextual information using probabilistic graphical models such as CRFs, our framework can robustly segment point cloud scenes into geometric surface classes. We validated our approach on multiple table setting datasets acquired in indoor environments with a laser sensor. The computational properties of our approach exhibit a favorable integration for fast, real time 3D classification of laser data.

As future work, we plan to increase the robustness of the FPFH descriptor against noise to be able to use it for point classification in datasets acquired using stereo cameras or other sensors less precised than laser scanners.

Acknowledgments This work was supported by the CoTeSys (Cognition for Technical Systems) cluster of excellence.

REFERENCES

[1] R. B. Rusu, Z. C. Marton, N. Blodow, and M. Beetz, "Learning Informative Point Classes for the Acquisition of Object Model Maps," in *Proceedings of the 10th International Conference on Control, Automation, Robotics and Vision (ICARCV), Hanoi, Vietnam*, 2008.

[2] M. Alexa and A. Adamson, "On Normals and Projection Operators for Surfaces defined by Point Sets," in *Proceedings of Symposium on Point-Based Graphics 04*, 2004, pp. 149–155.

[3] 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.

[4] G. Burel and H. Hénocq, "Three-dimensional invariants and their application to object recognition," *Signal Process.*, vol. 45, no. 1, pp. 1–22, 1995.

[5] N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann, "Robust Global Registration," in *Proc. Symp. Geom. Processing*, 2005.

[6] E. Akagunduz and I. Ulusoy, "3D Object Representation Using Transform and Scale Invariant 3D Features," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, 14-21 Oct.*, 2007.

[7] B.-C. M. and G. C., "Characterizing shape using conformal factors," in *Eurographics Workshop on 3D Object Retrieval*, 2008.

[8] 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.

[9] A. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, May 1999.

[10] F. Alex, D. Anthony, and H. A. van den, "Thrift: Local 3D Structure Recognition," in *DICTA '07: Proceedings of the 9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications*, 2007.

[11] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "SURF: Speeded Up Robust Features," in *Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346–359*, 2008.

[12] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[13] L. J. Skelly and S. Sclaroff, "Improved feature descriptors for 3-D surface matching," in *Proc. SPIE Conf. on Two- and Three-Dimensional Methods for Inspection and Metrology V*, 2007.

[14] D. Anguelov, B. Taskar, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, and A. Ng, "Discriminative learning of Markov random fields for segmentation of 3d scan data," in *In Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2005, pp. 169–176.

[15] R. Triebel, R. Schmidt, O. M. Mozos, , and W. Burgard, "Instace-based AMN classification for improved object recognition in 2d and 3d laser range data," in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), (Hyderabad, India)*, 2007.

[16] A. Barla, F. Odone, and A. Verri, "Histogram intersection kernel for image classification," in *Proceedings of International Conference on Image Processing (ICIP)*, 2003.

[17] J. Lafferty, A. Mccallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. 18th International Conf. on Machine Learning*. Morgan Kaufmann, San Francisco, CA, 2001, pp. 282–289.

[18] R. Klinger and K. Tomanek, "Classical probabilistic models and conditional random fields," Technische Universität Dortmund, Dortmund," Electronic Publication, 2007.

[19] E. H. Lim and D. Suter, "Conditional Random Field for 3D Point Clouds with Adaptive Data Reduction," *Cyberworlds, International Conference on*, vol. 0, pp. 404–408, 2007.