Object Class and Instance Recognition on RGB-D Data

Viktor Seib, Susanne Christ-Friedmann, Susanne Thierfelder, Dietrich Paulus
Active Vision Group (AGAS), University of Koblenz-Landau
Universitätsstr. 1, 56070 Koblenz, Germany
{vseib, scfriedmann, thierfelder, paulus}@uni-koblenz.de

ABSTRACT

We present a novel approach for combining 3D depth and visual information for object class and object instance recognition. Object classes are recognized by first assigning local geometric primitive labels using a CRF, followed by an SVM classification. Object instances are recognized using Hough-transform clustering of SURF features. Both algorithms perform well on publicly available object databases as well as on acquired data with an RGB-D camera. The object instance recognition algorithm was further evaluated during the RoboCup world championship 2012 in Mexico-City and won the first place in the Technical Challenge of the @Home-league.

Keywords: Conditional Random Field, Hough-clustering, Object Recognition

1. INTRODUCTION

Object recognition approaches solely based on RGB data allow to recognize object instances (e.g. *John's mug* vs. *Mary's mug*). However, these approaches fail when applied to objects with no texture or to unknown instances of an object class. Combining RGB and depth information not only allows to recognize previously learned object instances, but also provides information about geometric shapes of objects. Thus, even unknown object are classified as belonging to a certain class (e.g. *mug*, *can*).

We present an approach that follows the idea of [1], but extends it with a texture-based object recognition approach. The applied conditional random field (CRF) in our approach performs better on Fast Point Feature Histograms (FPFH) [2] than the CRF applied in [1]. Further, we use a modified version of Global Fast Point Feature Histograms (GFPFH) [1] for the support vector machine (SVM) classification that is faster to compute, but with equal accuracy. The RGB information of detected objects is passed to the 2D object recognition algorithm for object instance recognition. This approach is based on Hough-transform clustering of Speeded-Up Robust Features (SURF) [3] that allows reliable recognition even with partially occluded objects and heterogeneous backgrounds. Thus, the extension of the algorithm by 2D data not only allows for object class recognition, but also for object instance recognition. We tested our system with 6 different geometric primitive labels and 6 different object classes (in [1] 5 primitives and 4 objects classes were used).

In Sec. 2 we present related work in combining RGB and depth data for object recognition. Hereafter, an overview of the proposed image processing pipeline is given in Sec. 3. Sections 4 and 5 present the object class and instance recognition, respectively. Evaluation resulst are presented in Sec. 6 and Sec. 7 concludes the paper.

2. RELATED WORK

Recently, several approaches were proposed in order to incorporate 2D and 3D data for object recognition. In [4] every point is annotated with geometrical primitives. This local classification is used to compute a global descriptor for each detected object using an SVM. Additional processing is applied in order to calculate a grasping path for manipulation. Depth- and RGB data are combined to determine the exact position of objects in [5]. Four different features representing the texture and surface of the objects are extracted. Objects are classified using random forests. The algorithm presented in [6] requires several views from different angles of the scene. Data are acquired with a RGB-D camera and an Euclidean clustering algorithm is used to separate the points into objects. Additionally, SIFT-Features and HSV color histograms are extracted from image areas indicated by the objects. Objects are recognized by comparison with a database containing the extracted features.

Further author information: send correspondence to Viktor Seib, vseib@uni-koblenz.de

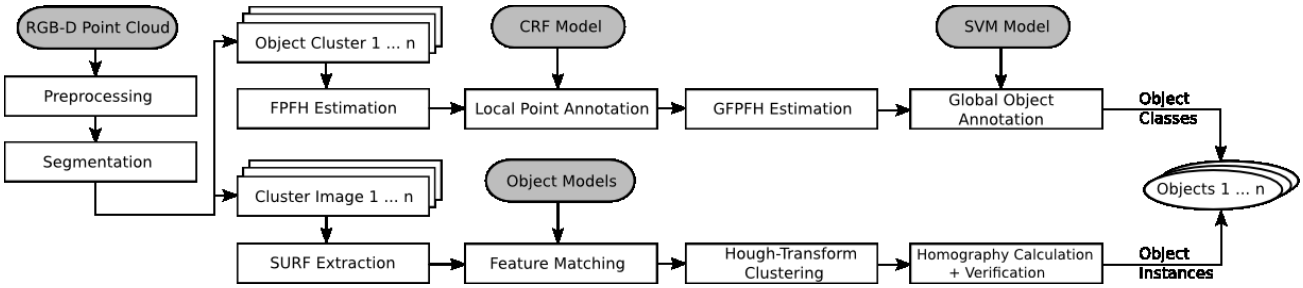

Figure 1. The proposed image processing pipeline. After a common preprocessing step the object recognition is split into object class recognition based on 3D data and object instance recognition based on object textures.

In contrast to these algorithms which combine 2D and 3D features, [7] and [8] propose a single descriptor that encompasses these data. Both approaches create multi-dimensional histograms combining color and geometry features and use SVMs for object classification.

The main purpose of the approach presented in [1] is to segment objects for mobile manipulation. Therefore, only 3D data for object classification is used in this two stages algorithm. The approach presented here extends this two stage object classification idea by additionally incorporating SURF from RGB-data, thus allowing for object class and object instance recognition.

3. IMAGE PROCESSING PIPELINE

The input for the image processing pipeline (Fig. 1) is an RGB-D point cloud. The output of the preprocessing are 3D object clusters and 2D cluster images that serve as input for object class and object instance recognition, respectively. At the end of the pipeline object classes and instances are combined to form the recognition results. Object class recognition always yields an output, whereas the instance recognition can output *object unknown*.

For the preprocessing different filters from the Point Cloud Library (PCL) [9] were used. We apply the *PassThrough* filter and subsequently the *UniformSampling* filter (voxel size of 6 mm) to reduce the input data size. Then, surface normals are computed and serve as input for the plane detection algorithm (here *SACSegmentationFromNormals*). An estimation of points above the table plane is obtained with the *ConvexHull* and *ExtractPolygonPrismData* filters. Finally, the *EuclideanClusterExtraction* filter is applied to obtain a separate point cloud for each of the objects and corresponding RGB-data for each object is extracted.

4. OBJECT CLASS RECOGNITION

The object class recognition algorithm is presented in this Section. First, the FPFH is calculated for each point in the dataset and labeled by the CRF with a geometric primitive type. The labeled points are subsequently passed to an SVM that determines a GFPFH for each object. Based on these global features the SVM assigns an object class to each object. We use a modified version of the GFPFH [4]. In contrast to the original implementation [1] no statistical label distribution is calculated in the modified version.

4.1 Conditional Random Field Model

A CRF is a discriminative graphical model that can be used for data labeling. It models the conditional probability $P(Y|X)$, where X is the set of conditional variables over a data sequence that have to be classified. Y is the set of conditional variables over the corresponding label sequence. Here, X consists of the FPFH descriptors of the points in the input data. The goal is to assign a label $y \in Val(Y)$ to each $x \in X$, where $Val(Y) = \{cylindric\_concave, cylindric\_convex,\ plane, edge, sphere, torus\}$ is the set of geometric primitive labels (second column in Fig. 2).

In our approach we use a pairwise CRF model based on [10]. This type of CRF model assigns a label y_i to an observed point x_i based on its feature descriptor (here: FPFH) as well as on $y_j \in Y_j$, where Y_j are the labels of the neighboring points of y_i in the constructed graph. The CRF is defined as

$$P(Y|X) = \frac{1}{Z(X)} \exp\left(\sum_{i \in S} \omega_l^T \Phi(f_i(X)) + \sum_{i \in S} \sum_{j \in MB_i} \nu_{l_i, l_{MB_i}}^T \Phi(f_{ij}(X))\right) \quad (1)$$

where $Z(X)$ is the partition function, $i, j \in S$ are nodes in the graph and S the set of all nodes. MB_i denotes the Markov-Blanket of node i (we use a fixed number of 8 neighbors per point). Based on the node specific parameter vector ω determined in the training phase, the association potential $\Phi(f_i(X))$ assigns the most probable label y_i to the observed point x given the calculated node features of all points X. The interaction potential $\Phi(f_{ij}(X))$ includes the labels of neighboring nodes into the classification process of x, penalizing different neighboring labels. The classification is based on the edge features and an edge specific parameter vector ν, acquired in the training phase. In our approach, node features are the aforementioned FPFHs and edge features are the angles, α, θ, ϕ, derived from the FPFH calculation which are determined pairwise between two neighboring points.

As a discriminative learning algorithm the CRF learns parameters on fully known datasets. For this purpose 3D point clouds have to be annotated by hand with the desired labels. We train the CRF using pseudo log-likelihood training with an optimization by the L-BFGS algorithm [11]. In the classification step, most probable labels Y need to be assigned to input data on which X was calculated. We convert our CRF graph into a cluster graph as described in [12] and use the loopy belief propagation algorithm [13] for classification.

4.2 Global Object Annotation with an SVM

In our approach we use the SVM implementation from libSVM [14]. The labeling output of the CRF on which the modified GFPFH is computed is applied for SVM training. Subsequently, objects are annotated by hand with one of the following six object categories: open cylinder, closed cylinder, box, mug, spherical object, bowl. In the classification step, GFPFH features are computed and the corresponding object class is inferred by the SVM.

5. OBJECT INSTANCE RECOGNITION

This Section describes the object instance recognition approach based on 2D images.

5.1 Training

In order to train the object recognition classificator an image of the background and of the object has to be captured. From this two images the difference is computed to separate the desired object from the background. SURF features are extracted from the acquired object image and stored in an object database. Further, images with a different object view can be acquired and added to the object model in the database. Bay et al. [3] recommend 30° as an optimal rotation between subsequently acquired images of an object for SURF computation.

5.2 Recognition

During object recognition no information about the background is available. Thus, SURF features are computed on the whole input image. For each feature in the input image the best feature in the database is obtained by nearest neighbor matching of the feature descriptor. In this step not all features can be matched e.g. because they do not appear in the database. On the other hand, some erroneous correspondences are obtained since each feature is considered individually and not in conjunction with the whole object.

These erroneous correspondences are discarded by applying Hough-transform clustering to find a consistent object pose. The idea is to create a multi dimensional accumulator with bins that represent object poses. Each feature correspondence is a hypothesis for an object pose and is added into the corresponding bin in the accumulator. Clusters of maxima in the Hough space correspond to most probable object poses, whereas bins with erroneous object poses get only little votes. Thus, outliers are removed and correct object poses are obtained. We use a 4 dimensional accumulator space to represent object transformations: 2 dimensions to encode translations in x and y direction, one to represent rotation and one for scale of the features. As suggested in [15]

Table 1. Accuracy of assigned geometric primitive labels by the CRF.

Geometric Primitive	Accuracy
Plane	92.4 %
Edge	79.7 %
Torus	77.2 %
Sphere	93.6 %
Cylinder convex	94.8 %
Cylinder concave	97.7 %
Overall Accuracy	92.8 %

Table 2. Accuracy of assigned primitive geometric labels per object class.

Object Class	Accuracy	Involved Primitives
Mug	93.7 %	convex, concave, torus
Closed Cylinder	88.4 %	convex, plane, edge
Open Cylinder	96.1 %	convex, concave
Bowl	73.2 %	convex, concave, plane
Box	95.8 %	plane, edge
Overall Accuracy	90.2 %	

and [16], to reduce quantification errors, the 2 closest bins in each dimension are added as a hypothesis, thus resulting in 16 accumulator entries per feature correspondence.

All bins containing at least 5 entries are considered as maxima in Hough space and used to find the best object pose. A perspective transformation is calculated between the features of a bin and the corresponding points in the database under the assumption that all features lie on a 2D plane. As most outliers were removed by the Hough-clustering a consistent transformation is obtained here. Using RANSAC, a homography is calculated for each bin and the bin resulting in a homography with the most point correspondences is considered to be the correct object pose. With the homography the recognized object is projected into the scene. To speed up the algorithm all bins are sorted in descending order considering the number of entries. A homography is calculated starting with the bin containing the highest number of features. The calculation terminates if the next bin contains less features than the number of found point correspondences in the previous homography.

Finally, the results are verified by calculating the object presence probability $p = \frac{f_m}{f_t}$, where f_m is the number of matched features of an object and f_t is the total number of features that are present in the area of the same object. The number of features in the object area is calculated by projecting the object into the scene using the calculated homography and then counting all features in the bounding box of the projected object. A detailed description of this SURF based object recognition approach is given in [17].

6. EXPERIMENTS AND RESULTS

This section presents experiments and results of the algorithms discussed above. Experiments were performed on objects from the database presented in [18] and also on data acquired from typical household items. The object instance recognition approach was evaluated on test data from the database presented in [19]. All images in this database have a resolution of 640×480 pixels.

6.1 Object Class Recognition

To test the accuracy of the CRF 58 different objects belonging to 12 different categories from the database in [18] were selected and annotated with geometric primitive labels. From these 58 objects, 26 were used for training. Among the selected categories were mugs, bowls, boxes, cans, balls, and fruits. The results of the point labeling with geometric primitive are presented in Tab. 1. The same results with relation to the object classes that were used for testing are presented in Tab. 2. The accuracy was determined as the ratio of correctly labeled points by the CRF and all points with the corresponding label in the manually annotated data.

The accuracy for most geometric primitives is close to or above 90 % (Tab. 1). The classes *edge* and *torus* have the lowest accuracy in the tests of slightly below 80 %. In the case of *torus* many points were classified as having concave or convex geometry. Many points that were labeled as *edge* were classified as plane geometry. Errors partially stem from the manual labeling process, as it can be hard to distinguish between e.g. edge and plane or sphere and convex cylinder in complex object classes. In Tab. 2 the CRF labeling results are grouped according to the six object classes that we want to recognize. Again, most results are close to or above 90 %.

If the manual labeling is done accurately and the objects chosen for training contain a balanced representation of different geometric types, the classification will perform well also on unknown objects of the known categories as the results in Tab. 1 suggest.

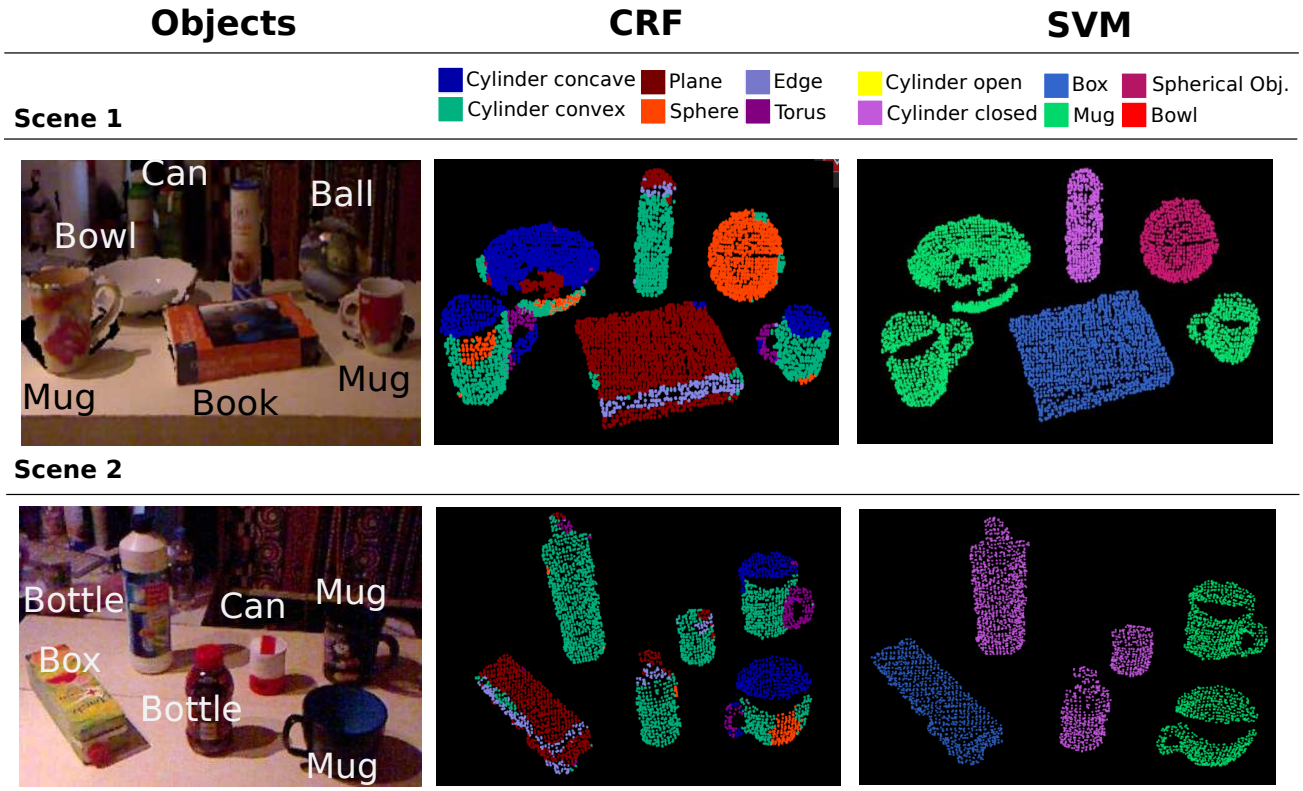

Figure 2. The left column shows the original image with annotated objects. The middle and right column show classification results of the CRF and the SVM, respectively. The colors indicate the assigned labels and object classes.

The results as presented in Tab. 2 were used to train the SVM. The training time of the SVM took approximately 3 s per object. However, the classification takes about 30 s per object, most of the time is consumed for the GFPFH calculation.

The whole object class recognition pipeline was evaluated on typical object in a household environment. Example classification results are depicted in Fig. 2. A total of 30 objects was used for evaluation, no new training was performed on these objects. Data were acquired with a RGB-D camera placed in a height of 1.4 m and 0.8 m in front of a table containing the objects. The SVM assigned 24 of 30 objects (80 %) the correct object class type. The reasons for the classification errors are that all of the misclassified objects contain many points in areas with concave and convex geometry. The differences between the classes are rather small: mugs have some points on the handle, whereas bowls have some points in the planer base. As these small differences are not sufficiently reflected in the GFPFH, the SVM assigned wrong object classes.

6.2 Object Instance Recognition

Experiments were performed to test the influence of the accumulator size, variable backgrounds and light conditions as well as partial occlusion on the performance of the classification. The algorithm was trained with 5 different objects and 5 views per object. The classification was performed on the same 5 objects, but with 10 different views per object. For the verification step we used a threshold of 15 % object presence probability and a minimum of 5 matched features per object.

Our experiments show that the instance recognition works best with an angle difference of 30° or less between object views. The size of each dimension of the accumulator is a crucial parameter for the performance of the algorithm. In our approach 10 bins per dimension proved to be a good trade-off between quantification errors (if too many bins are used) and insufficient accuracy (if too little bins are used).

Table 3. Object recognition results on images with different backgrounds. The numbers in brackets indicate the number of false positive recognitions.

Object	hom. back.	weak het. back.	strong het. back.
bscup	100 %	90 % (1)	100 %
nizoral	100 %	100 % (2)	90 %
perrier	100 %	100 % (1)	100 % (2)
ricola	100 %	100 % (1)	100 %
truck	100 %	90 %	70 % (1)

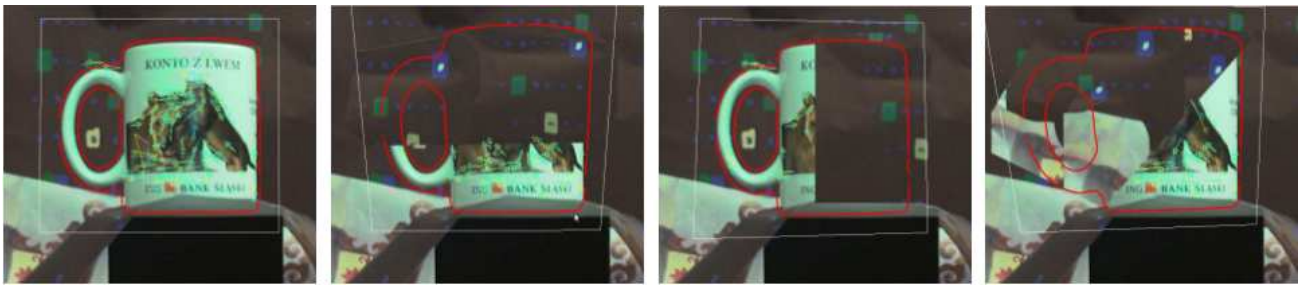

Figure 3. Example images for detection of partially occluded objects. From left to right: the unoccluded object is recognized with 98 matched features and 38 % confidence. The occluded object have less features, but are still recognized correctly: 49 with 33 %, 17 with 17 %, and 34 with 34 %.

The employed database contains images of the same objects with homogeneous, weak heterogeneous, and strong heterogeneous backgrounds. Different light conditions are used in the images with non-homogeneous backgrounds. The experimental results of our algorithm with different backgrounds are presented in Tab. 3.

Another experiment was performed to test the algorithm with partially occluded objects. Occlusion was simulated by partially replacing the object in the test data with the corresponding background. The results are presented in Fig. 3. The unoccluded object is recognized with a total of 98 matched features and an object presence probability of 38 %. With increasing occlusion the number of features decreases, but is still high enough for a correct recognition of the object. However, with increasing occlusion the accuracy of the computed homography (red lines in Fig. 3) and thus of the bounding box decreases. A further evaluation and comparison with a statistic object recognition approach is presented in [20].

This object instance recognition approach was also applied during the Technical Challenge in the @Home-league of the RoboCup that took place in Mexico-City in 2012. Over 50 objects were placed on a table containing randomly selected 15 of 25 previously known objects. With this approach our robot could correctly identify 12 of the 15 known objects correctly, while at the same time having no false positive recognitions. With this result our robot places first in the Technical Challenge. The input image for object recognition as well as the recognition results are shown in Fig. 4.

7. CONCLUSIONS AND FUTURE WORK

We presented a combined object class and object instance recognition approach. Object class recognition is performed in two steps. First, a CRF annotates points with geometric primitives labels with an accuracy of 92.8 %. This is a higher accuracy than in [1] with FPFH. However, with a modified FPFH a better accuracy is achieved in [1]. We expect that this modified features could also improve our classification result. Unfortunately, this modified FPFH is not available in the Point Cloud Library, yet. After the CRF classification, an SVM assigns an object class based on the GFPFH. The overall classification result achieved after the SVM is 80 %.

The proposed object instance recognition approach performs well on different object poses and achieves good results on scenes with heterogeneous backgrounds and partially occluded objects. This was also proved by the outstanding performance of our robot in the Technical Challenge during the RoboCup 2012 in Mexico-City.

Our future work will concentrate on improving both recognition algorithms. Since our CRF provides good results, we will improve our object class recognition by applying a hierarchical CRF for local and global object

Figure 4. The input image for object recognition as acquired by our robot during the Technical Challenge of the RoboCup (left). Object recognition results with 12 correctly identified object instances (right).

annotation instead of using a combination of CRF and SVM. Further, we plan to replace the GFPFH by a different feature since GFPFH is slow to compute and does not generalize well the overall object shape. Future improvements in the object instance recognition include color classification for featureless object.

REFERENCES

[1] R. B. Rusu, A. Holzbach, M. Beetz, G. Bradski, Detecting and segmenting objects for mobile manipulation. Proc. of ICCV Workshops, (2009).

[2] R. B. Rusu, N. Blodow, M. Beetz, Fast Point Feature Histograms (FPFH) for 3D registration. Proc. of ICRA, (2009).

[3] H. Bay, T. Tuytelaars, L. V. Gool, SURF: Speeded Up Robust Features. Proc. of ECCV, Springer, (2006).

[4] Z.-C. Marton, D. Pangergic, R. B. Rusu, A. Holzbach, M. Beetz, Hierarchical Object Geometric Categorization and Appearance Classification for Mobile Manipulation. Proc. of International Conference on Humanoid Robots, (2010).

[5] J. Stuckler, S. Behnke, Combining depth and color cues for scale- and viewpoint-invariant object segmentation and recognition using random forests. Proc. of IROS, (2010).

[6] J. Tang, S. Miller, A. Singh, P. Abbeel, A Textured Object Recognition Pipeline for Color and Depth Image Data. ICRA Solutions in Perception Challenge, (2011).

[7] A. Kanezaki, Z.-C. Marton, D. Pangercic, T. Harada, Y. Kuniyoshi, M. Beetz, Voxelized Shape and Color Histograms for RGB-D. Proc. of IROS Workshops, (2011).

[8] B. Browatzki, J. Fischer, B. Graf, H. H. Bulthoff, C. Wallraven, Going into depth: Evaluating 2D and 3D cues for object classification on a new, large-scale object dataset. Proc. of ICCV Workshops, (2011).

[9] R.B. Rusu, S. Cousins, 3d is here: Point Cloud Library (pcl), http://pointclouds.org/. ICRA, (2011).

[10] J. Niemeyer, C. Mallet, F. Pereira, U. Sörgel, Conditional Random Fields for the Classification of LiDAR Point Clouds. Proc. of ISPRS Workshop, (2011).

[11] N. Okazaki. libLBFGS: A Library of Limited-Memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS), http://www.chokkan.org/software/liblbfgs, 2010.

[12] D. Koller, N. Friedman, Probabilistic Graphical Models: Principles and Techniques. The MIT Press, (2009).

[13] K. P. Murphy, Y. Weiss, M. I. Jordan, Loopy Belief Propagation for Approximate Inference: An Empirical Study. Proc. of UAI, (1999).

[14] C. C. Chang, C. J. Lin, LIBSVM: a library for support vector machines. ACM TIST, vol. 2, (2011).

[15] D. G. Lowe, Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, Springer, (2004).

[16] W. E. L. Grimson, D. P. Huttenlocher, On the sensitivity of the Hough transform for object recognition. Transactions on Pattern Analysis and Machine Intelligence, (1990).

[17] V. Seib, M. Kusenbach, S. Thierfelder, D. Paulus, Object Recognition Using Hough-Transform Clustering of SURF Features, http://www.ros.org/wiki/obj_rec_surf, 2013.

[18] K. Lai, L. Bo, X. Ren, D. Fox, A large-scale hierarchical multi-view rgb-d object dataset. ICRA, (2011).

[19] M. Grzegorzek, H. Niemann, Statistical object recognition including color modeling. Image Analysis and Recognition, Springer, (2005).

[20] P. Decker, S. Thierfelder, D. Paulus, M. Grzegorzek, Dense Statistic Versus Sparse Feature-Based Approach for 3D Object Recognition. PRIA 2010. Moscow: Springer MAIK Nauka/Interperiodica, (2010).