Recognizing Point Clouds using Conditional Random Fields

Farzad Husain*, †Babette Dellen and *Carme Torras
*Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Llorens i Artigas 4-6, 08028, Barcelona, Spain
†RheinAhrCampus der Hochschule Koblenz, Joseph-Rovan-Allee 2, 53424 Remagen, Germany
Email: shusain@iri.upc.edu, dellen@hs-koblenz.de, torras@iri.upc.edu

***Abstract*—Detecting objects in cluttered scenes is a necessary step for many robotic tasks and facilitates the interaction of the robot with its environment. Because of the availability of efficient 3D sensing devices as the Kinect, methods for the recognition of objects in 3D point clouds have gained importance during the last years. In this paper, we propose a new supervised learning approach for the recognition of objects from 3D point clouds using Conditional Random Fields, a type of discriminative, undirected probabilistic graphical model. The various features and contextual relations of the objects are described by the potential functions in the graph. Our method allows for learning and inference from unorganized point clouds of arbitrary sizes and shows significant benefit in terms of computational speed during prediction when compared to a state-of-the-art approach based on constrained optimization.**

I. Introduction

Range sensing devices using active illumination for depth estimation such as Microsoft Kinect and LIDAR provide a 3D representation of the scene in form of a point cloud which can be used for robot perception. In this context, the semantic labeling of cluttered indoor scenes is highly important for facilitating the interaction of the robot with its environment, for example in mobile robotics or robot manipulation, where objects have to be targeted and grasped. An example of a labeled point cloud according to common object categories for an office environment is shown in Figure 1. The goal of semantic labeling is to assign object labels, such as "monitor", "table", or "wall" to the respective parts of the scene. Commonly, the data is first subdivided into parts, then a 3D graphical model is constructed that captures the features and contextual relations of the parts, and used for learning and recognition.

In the past, several methods have been proposed for learning the semantics of objects that exploit contextual information in color images for object recognition [1], [2]. One problem with color images is that they lack the necessary discriminative properties of the underlying 3D geometry. To overcome this limitation, some approaches extracted first 3D information from the images using stereo [3] or monocular depth cues [4], but due to the limited accuracy of passive methods, this did not lead to a significant improvement, as recently pointed out in [5].

Using range data from active sensing for semantic labeling has been shown to improve object recognition [5], [6]. Here, many approaches use the local arrangement of individual object parts with respect to each other for object recognition [7], [6]. But besides these local contextual cues describing the object, its global, coordinate invariant geometric relations with other objects provide additional important clues for its recognition [8]. Graphical models provide a unified framework that allows incorporating both the object's local and global features and have for this reason been widely used for this task [5], [9], [10]. However, current approaches are rather time-expensive which makes them unfeasible for online classification. To reduce the computation times for these models, inherent constraints have to be relaxed leading to a decrease in accuracy.

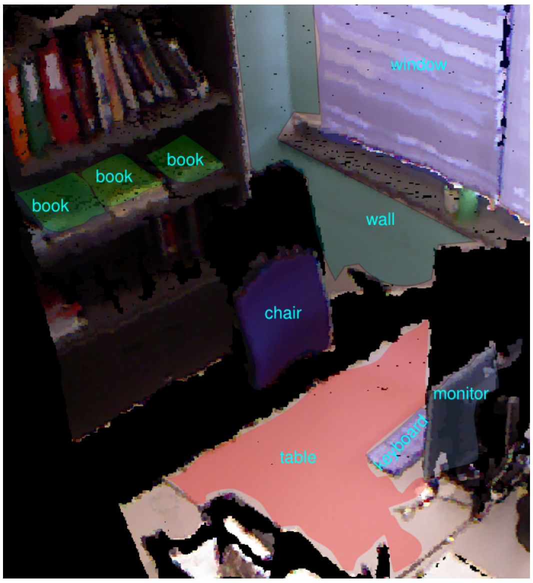

Fig. 1. An example of a labeled point cloud of a typical office environment.

In this work, we explore the use of Conditional Random Fields (CRFs) for time-efficient approximate inference. Similar to previous approaches, we use a graphical model to describe the point clouds and encode features and relations of point-cloud segments, representing surface patches. We choose CRFs over the more commonly used Markov Random Fields because it is a more direct approach for modeling the probability of labels [11]. This leads to a significant improvement in terms of computation time while yielding predictions with an accuracy similar to the ones obtained with exact inference [5]. We also propose the use of Point Feature Histogram (PFH) [12] as a 3D shape descriptor of point cloud segments. This descriptor has recently been shown to outperform others for recognition tasks in terms of precision [13]. We tested the Conditional

Random Field framework on challenging RGB-D and range-only datasets. We evaluated our approach on three different kinds of scenes and compared its performance to a state-of-the-art method [5].

II. Related Work

A large body of work has been conducted in the area of object recognition from 2D images in the past. We can only provide a rough overview here and focus on those methods that use undirected graphs for image representation [2], [14], [11], [15], [16], [17], [18].

In [2], object recognition is performed by constructing a graph using different parts of an object together with their relative arrangements. Similar part-based models for object recognition are adopted in [14]. Using relative arrangements of different objects within a scene for recognition has also been performed in [11], [15], [16], [17]. In [11], [17], regional and global image features are incorporated in a CRF. It is also argued in [11], [2] that when the goal is to estimate only the posterior over the labels given the input data (point cloud in our case), a discriminative model (CRF) is more suitable when compared to a generative one (Markov Random Field). In [18], objects are relabeled according to their contextual relevance in a post-processing step inside a CRF framework.

Compared to 2D images, far less work has been done for 3D scenes [19], [6], [20], [5]. In [19], Associative Markov Network models are learned using a functional gradient technique for labeling point clouds of outdoor scenes acquired using a LIDAR sensor. Object detection after learning from its multiple isolated views is shown in [6]. Labeling of planar patches in a point cloud using context is performed in [20].

Our work is closely related to [5], where each segment of the input point cloud is represented by a node in a graph and the relational information between different segments is modeled using pairwise edge potentials. In order to make the model suitable for on-line classification, the constraints in the proposed optimization problem were relaxed which led to a drop in recall. In comparison, our time-efficient approximate inference model yielded results similar to the exact inference as in [5].

III. Problem Formulation

Given a set X of n segments, which is a subset of an over-segmented, input point cloud $\mathcal{X}$, i.e., $X = \{x_1, x_2, \ldots, x_n\} \subseteq \mathcal{X}$, the goal is to determine a set of unique semantic labels $Y = \{y_1, y_2, \ldots, y_n\}$ for each segment. Each y_i is a member of the output set $\mathcal{Y}$. Here, $\mathcal{Y}$ is a finite set of k object categories that frequently occur in a particular scene. The segment x_i is essentially a vector of variable length containing information about the position of the sampled points in Euclidean space which can also contain color information.

IV. Methodology

Our approach begins with the construction of a graphical model of the given point cloud, which is segmented by forming clusters based on the differences in the local surface normals and the connectivity of the surfaces. Each point cloud is represented as a set of nodes and edges, and their potential functions are constructed (Section IV-A). During learning (Section IV-C), we estimate the node and edge weights that maximize the conditional likelihood of the labels given the input features (Section IV-D). We use the same approximate inference method for both the prediction of labels and during learning (Section IV-B).

A. Graphical Model

We use a graphical structure G analogous to [5], i.e., each segment x_i in the point cloud is represented by a node which can have exactly one label. An edge exists between two nodes if a distance measure between the corresponding segments is less than a threshold. We define the potential function $\Psi(Y, X; w)$ over unary and pairwise cliques, i.e.,

$$\begin{aligned}\Psi(Y, X; w) = & \sum_{i \in \mathcal{N}} \sum_{l} u_l^{y_i} \cdot \phi_l(x_i) \\ & + \sum_{(i,j) \in \mathcal{E}} \sum_{m} v_m^{y_i y_j} \cdot \varphi_m(x_i, x_j), \end{aligned} \tag{1}$$

where $\mathcal{N}$ is the set of nodes and $\mathcal{E}$ is the set of edges. The parameters $u_l^{y_i}, v_m^{y_i y_j}$ are the components of the parameter vector w. Each node x_i and edge (x_i, x_j) is represented by a discriminative node feature vector $\phi_l \in \mathbb{R}^{d_1}$ and an edge feature vector $\varphi_m \in \mathbb{R}^{d_2}$, respectively. The products $u_l^{y_i} \cdot \phi_l(x_i)$ and $v_m^{y_i y_j} \cdot \varphi_m(x_i, x_j)$ determine the discriminative strength of each node feature ϕ_l for the label y_i and edge feature φ_m for labels (y_i, y_j), respectively.

Unlike [5], where the parameter w is optimized for the joint probability distribution $P(X, Y; w)$, we optimize w using the maximum likelihood for the conditional distribution $P(Y|X; w)$ which we define as

$$P(Y|X, w) = \frac{e^{\Psi(Y,X;w)}}{\sum\limits_{y \in \mathcal{Y}} e^{\Psi(Y,X;w)}}, \tag{2}$$

The parameter w for optimizing the conditional likelihood (Equation 2) is learnt from the training examples which will be explained in Section IV-C.

B. Inference

In order to predict the object category labels, given a set of input segments X and a learned parameter vector w^*, we will choose the labels that give the maximum a posteriori (MAP) labeling of the conditional distribution, i.e.,

$$\hat{Y} = \arg \max_{Y \in \mathcal{Y}^n} P(Y|X; w^*). \tag{3}$$

Our model assumes a large number of object classes that considering all labels, leads to a computational complexity of $\mathrm{O}(\mathcal{Y}^n)$ which makes exact inference intractable. Hence we use loopy belief propagation (LBP) [21], [22] for approximate inference which has been shown to approximate the log-likelihood better than other methods, e.g., mean field approximation [21].

C. Learning

For learning the parameter vector w, we use the L2-regularized, conditional log-likelihood as the loss function [23], i.e.,

$$\mathcal{L}(w) = \lambda||w||^2 - \sum_{i=1}^{p} \log P(Y_i|X_i;w), \tag{4}$$

where p is the number of training examples and λ is the regularization parameter that prevents over-fitting the parameter w on the training data. To estimate the parameter vector w, i.e., $w^* = \arg\min_w \mathcal{L}(w)$, we use mini-batch stochastic gradient descent (SGD) [24]. In mini-batch SGD, the parameter vector w is updated after every iteration j according to

$$w_{j+1} = w_j - \frac{\eta_j}{b}\sum_{i=1}^{b}\frac{\partial}{\partial w}\mathcal{L}_i(w_j), \tag{5}$$

where b is the batch size that we determine empirically by cross-validation on the training data. We use the decreasing step size η_j at each iteration j as proposed in [23], i.e.,

$$\eta_j = \frac{\eta_0}{1 + j/p}, \tag{6}$$

where η_0 is a constant. In order to solve Equation 5, we need to determine the gradient of the loss function at each iteration j, i.e.,

$$\frac{\partial}{\partial w}\mathcal{L}_i(w) = 2\lambda w - \frac{\partial}{\partial w}\log\left(\frac{e^{\Psi(Y_i,X_i;w)}}{\sum\limits_{y\in\mathcal{Y}} e^{\Psi(Y_i,X_i;w)}}\right) \tag{7}$$

We first differentiate Equation 7 with respect to the parameters u_l, corresponding to the node features (see Equation 1), for a training example i, i.e.,

$$\frac{\partial}{\partial u_l}\mathcal{L}_i(w) = 2\lambda u_l + \sum_{j\in\mathcal{N}}\phi_l(x_j) - \sum_{y\in\mathcal{Y}}P(Y|X;w)\sum_{j\in\mathcal{N}}\phi_l(x_j). \tag{8}$$

Note that in Equation 8 the segment x and label y belong to the i-th training example. The gradient computed is simply the regularization parameter plus the difference of a feature from its expected value. The expected value of the feature, i.e., $\sum\limits_{y\in\mathcal{Y}} P(Y|X;w) \sum\limits_{j\in\mathcal{N}} \phi_l(x_j)$, is calculated using loopy belief propagation. Similarly, we can differentiate Equation 7 with respect to the edge parameters v_m as follows

$$\frac{\partial}{\partial v_m}\mathcal{L}_i(w) = 2\lambda v_m + \sum_{(j,k)\in\mathcal{E}}\varphi_m(x_j,x_k) - \sum_{y\in\mathcal{Y}}P(Y|X;w)\sum_{(j,k)\in\mathcal{E}}\varphi_m(x_j,x_k). \tag{9}$$

D. Features

The choice of node and edge features mainly depends on the nature of the acquired data. We tested our model on data from two different kinds of range sensors, i.e., Microsoft Kinect (short range, for indoor scenes along with color) and LIDAR (long range, for outdoor scenes).

TABLE I. Node features computed for each segment x_i for data from Kinect sensor

Feature	Count
Histogram of HSV color values	14
Average of HSV color values	3
Average of HOG features	31
Linearity	1
Planarity	1
Scatter	1
Vertical component of the normal	1
Vertical and horizontal extent of bounding box	2
Distance from the scene boundary	1

TABLE II. Edge features computed for segments x_i and x_j for data from Kinect sensor.

Feature	Count
Difference of avg HSV color values	3
Coplanarity and Convexity	2
Horizontal and vertical distance between centroids	2
Angle between surface normals	2
Distance between closest points	1
Relative position from camera	1

TABLE III. Node features computed for each segment x_i for data from LIDAR sensor.

Feature	Count
Linearity	1
Planarity	1
Scatter	1
Volume of Convex Hull	1
PFH	27

1) Features for Kinect sensor: Table I and Table II provide a list of the node and edge features, respectively, for data from the Kinect sensor. A detailed description of how these features are computed can be found in [5].

2) Features for LIDAR sensor: Table III provides a list of the node features that we used for data acquired with the LIDAR sensor. The spectral features for linearity, planarity and scatter are the same as the ones used in [19], [25]. In addition we also used the Point Feature Histogram (PFH) and the volume of the convex hull V of each segment. PFH is originally computed by determining 4 angle relations between every pair of points in a k-neighborhood, where the neighborhood is usually a sphere with a fixed radius. Different from them, we define this neighborhood as all the 3D points belonging to a segment x_i. Afterwards, all points are binned in a 27-dimensional histogram. We use the concatenation of the spectral features of both nodes x_i and x_j as edge features.

E. Cumulative Binning

All the features are binned using a cumulative binning strategy [5], i.e., each feature is represented by $n_b = 10$ binary values. Instead of creating a single node potential for each node feature as in [5], we found that the precision increased after grouping two consecutive bins together and treating them as a single node potential. Hence, in total, the number of weights that we learn are $n_{\text{tot}} = (n_{\text{nodeFeatures}} \times (n_b/2) \times n_{\text{states}}) + (n_{\text{edgeFeatures}} \times n_{\text{states}} \times n_{\text{states}})$, where $n_{\text{nodeFeatures}}$ is the number of node features, n_{states} is the number of labels and $n_{\text{edgeFeatures}}$ is the number of edge features.

V. Experiments

We tested our model on the publicly available Cornell rgb-d datasets[1] (CRGBD) [5] for indoor scenes and the Oakland 3-D Point Cloud dataset[2] (OPCD) [19] for outdoor scenes. In CRGBD, one dataset consists of home scenes and another of office scenes acquired with the Microsoft Kinect sensor, along with a human annotation. The OPCD dataset contains outdoor scenes acquired with a LIDAR sensor along with a human annotation. We evaluate our model separately on each of these scenes.

A. Evaluation measure

For evaluating the prediction accuracy, we use the precision and recall metric commonly found in the pattern recognition literature [5], [26]. The precision value p_c of the c-th class is defined as $p_c = t_c/i_c$, where t_c is the number of correctly classified segments (true positives) and i_c is the number of segments predicted as the c-th class. The recall value r_c is defined as $r_c = t_c/n_c$, where n_c is the number of ground-truth segments in the c-th class. In order to aggregate the precision and recall over all the k object categories, we calculate both the micro and macro-average precision/recall [5]. The micro average gives an average of the precision/recall over the number of samples, whereas the macro average is the average of the individual precision/recall of each category. Hence, in the case of micro average, categories having more samples are given more importance.

B. Results for CRGBD

We first conducted experiments on the home and office scenes from CRGBD. A graphical model is constructed including a set of nodes and edges for each example. Each segment within the point cloud is used as a node and an edge is drawn between two nodes if the minimum distance between the corresponding segments is less than *context_range* as defined in [5]. In order to compare our CRF model to the MRF model of [5], we used the same node and edge potentials (associative and non-associative) as in [5]. The examples are divided into training and test sets according to the 4-fold cross validation. We learn the weights using the training set (see Section IV-C) and then predict the labels for the test set using the learned model (see Section IV-B).

Table IV shows a comparison of our results with the ones obtained in [5], using the micro and macro-average precision/recall for CRGBD. Here, *crf_node_only* refers to our model without taking the contextual relations into account which can be interpreted as a multinomial logistic-regression model showing the effect of node features only. The term *crf_lbp* refers to the full model with both node and edge potentials. We obtained very similar results to the multi-class SVM (*svm_node_only*) [5]. A small drop in precision is observed when comparing approximate inference under the CRF framework (*crf_lbp*) with exact inference under the MRF one (*svm_mrf_parsimon*) [5].

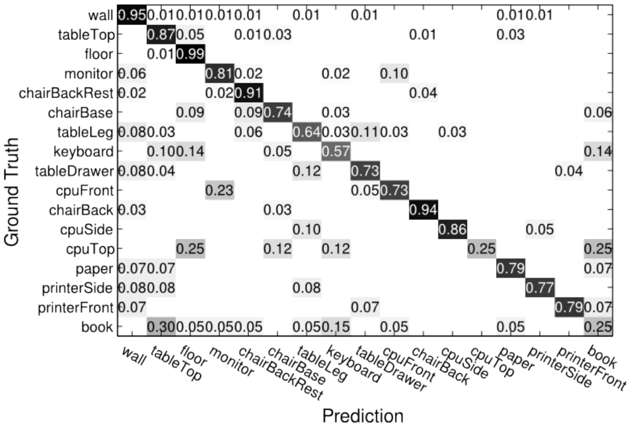

Fig. 2. Confusion Matrix of our results for the office dataset from CRGBD [5] with 17 categories.

Our approach leads to a considerable increase in computational speed, i.e., ∼0.014 seconds, on a single core implementation in C++. In [5], the computational time for exact inference (∼18 minutes) was reduced to ∼0.05 seconds by relaxing the constraints in the proposed optimization problem, but this led to a significant drop in recall. In comparison, our model permits time-efficient approximate inference with a smaller decrease in precision and recall when tested on the same dataset along with the same potentials as in [5]. This makes our model more suitable for on-line classification.

Figure 2 and 3 show the normalized confusion matrices for the office and home datasets from CRGBD, respectively. Our model provides correct predictions for most object categories as indicated by the large values on the diagonals compared to non-diagonal entries of the matrices. In the office dataset, some categories are more easily confused than others, for example, the book lying on a table can get confused with the tableTop (Fig. 2). Similar problems are observed in the home dataset, where the books can get confused with the shelfRack (Fig. 3).

We made similar observation about the confusion matrices as reported in [5], i.e., the object categories in the office scenes were less confused when compared to the home scenes. We also noticed that some of the categories such as the book and table drawer are less confused with our approach.

C. Results for OPCD

Next, we conducted experiments on scenes from OPCD. As before, a graphical model is constructed for each example, where each segment within the point cloud is used as a node. We define an edge between two nodes if the average distance of the corresponding segments is less than 5 m. Here, only the closest nearest neighbors are used, representing 20% of all neighbors. From the 17 examples in OPCD we selected 16 and divided them into training and test sets according to the 4-fold cross validation. We learned the weights using the training set

[1] Available:http://pr.cs.cornell.edu/sceneunderstanding/data/data.php

[2] Available:http://www.cs.cmu.edu/∼vmr/datasets/oakland_3d/cvpr09/doc/

TABLE IV. A COMPARISON OF AVERAGE MICRO PRECISION/RECALL, AND AVERAGE MACRO PRECISION AND RECALL FOR HOME AND OFFICE SCENES FROM CRGBD [5]. DATA FOR *svm_node_only* AND *svm_mrf_parsimon* TAKEN FROM TABLE II IN [5].

	Office Scenes			Home Scenes		
	micro	macro		micro	macro	
Algorithm	P/R	Precision	Recall	P/R	Precision	Recall
svm_node_only [5]	77.97	69.44	66.23	56.50	37.18	34.73
crf_node_only	78.11	69.72	63.45	58.64	39.40	36.41
svm_mrf_parsimon [5]	84.06	80.52	72.64	73.38	56.81	54.80
crf_lbp	83.28	79.80	73.90	69.11	54.91	50.82

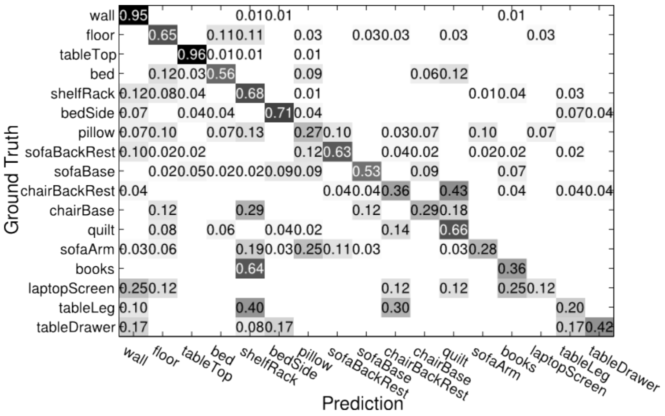

Fig. 3. Confusion Matrix of our results for the home dataset from CRGBD [5] with 17 categories.

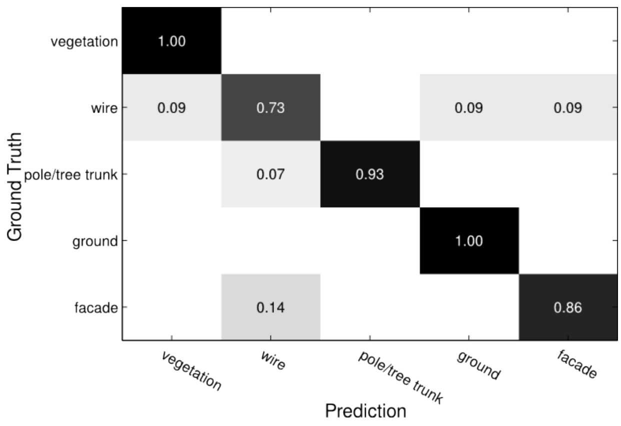

Fig. 4. Confusion Matrix of our results for OPCD [19] with 5 categories.

(see Section IV-C), and then predicted the labels for the test set (see Section IV-B).

Table V shows the evaluation results for OPCD. Here, *crf_node_only* refers to the model without taking the contextual relations into account, and *crf_lbp* refers to the model with both node and edge potentials. We obtained a considerable increase in precision after using PFH as an additional feature both in *crf_node_only* and *crf_lbp*.

Fig. 4 shows the normalized confusion matrix for OPCD. We have less confusion in this scenario. This is mainly due to the smaller number of categories in this dataset.

TABLE V. AVERAGE MICRO PRECISION/RECALL, AND AVERAGE MACRO PRECISION AND RECALL FOR OPCD [19].

Algorithm	Features	micro	macro	
		P/R	Precision	Recall
crf_node_only	spectral features,V	48.61	46.19	48.22
crf_node_only	spectral features,V, PFH	81.94	80.78	80.04
crf_lbp	spectral features,V	75.00	74.62	74.66
crf_lbp	spectral features,V, PFH	91.67	90.65	90.35

VI. CONCLUSION AND FUTURE WORK

We have explored the Conditional Random Field framework for modeling local features and contextual relations of objects from point clouds and tested them on datasets acquired with two different range sensing devices. We chose the mini-batch stochastic gradient descent for optimization, as it is well known for its faster convergence compared to other optimizers such as limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm [27]. We achieved a similar precision in the prediction of correct object labels as a state-of-the-art method based on a Markov Random Field framework [5] while improving computational speed.

We further showed that the inclusion of the Point Feature Histogram as a feature leads to a significant increase in precision for the OPCD dataset. We did not observe the same effect for the CRGBD. This is because the segments here have a smooth 3D shape for almost all object categories [5], as compared to [19], and thus the Point Feature Histogram does not provide any additional information about the 3D structure of the object.

In the future, we plan to extend our approach to dynamic scenes [28] and incorporate the motion parameters obtained from a tracker [29], during learning. Stochastic gradient descent will allow us to dynamically update the learned parameters for new examples. We also plan to use more advanced inference techniques such as convex belief propagation which guarantees convergence for graphical models with loops [30].

ACKNOWLEDGMENT

This work was supported by the EU project (IntellAct FP7-269959), the project PAU+ (DPI2011-27510), and the CSIC project CINNOVA (201150E088). B. Dellen was supported by the Spanish Ministry for Science and Innovation via a Ramon y Cajal fellowship.

REFERENCES

[1] M. J. Choi, A. Torralba, and A. Willsky, "A tree-based context model for object recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 2, pp. 240–252, 2012.

[2] A. Quattoni, M. Collins, and T. Darrell, "Conditional random fields for object recognition," in *Advances in Neural Information Processing Systems*. MIT Press, 2004, pp. 1097–1104.

[3] F. Tombari, F. Gori, and L. Di Stefano, "Evaluation of stereo algorithms for 3d object recognition," in *IEEE International Conference on Computer Vision Workshops*, 2011, pp. 990–997.

[4] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce, "3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints," *Int. J. Comput. Vision*, vol. 66, no. 3, pp. 231–259, 2006.

[5] A. Anand, H. S. Koppula, T. Joachims, and A. Saxena, "Contextually guided semantic labeling and search for three-dimensional point clouds," *The International Journal of Robotics Research*, vol. 32, no. 1, pp. 19–34, 2013.

[6] K. Lai, L. Bo, X. Ren, and D. Fox, "Detection-based object labeling in 3d scenes," in *IEEE International Conference on Robotics and Automation*, 2012, pp. 1330–1337.

[7] S. Savarese and L. Fei-Fei, "3d generic object categorization, localization and pose estimation," in *IEEE 11th International Conference on Computer Vision*, 2007, pp. 1–8.

[8] T. De Laet, S. Bellens, R. Smits, E. Aertbelien, H. Bruyninckx, and J. De Schutter, "Geometric relations between rigid bodies (part 1): Semantics for standardization," *IEEE Robotics Automation Magazine*, vol. 20, no. 1, pp. 84–93, 2013.

[9] K. Murphy, A. Torralba, and W. T. Freeman, "Using the forest to see the trees: a graphical model relating features, objects and scenes," in *Advances in Neural Information Processing Systems*, 2003.

[10] C. Li, A. Saxena, and T. Chen, "θ-mrf: Capturing spatial and semantic structure in the parameters for scene understanding," in *Advances in Neural Information Processing Systems*, 2011, pp. 549–557.

[11] X. He, R. Zemel, and M. Carreira-Perpindn, "Multiscale conditional random fields for image labeling," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2004, pp. 695–702.

[12] R. B. Rusu, Z. C. Marton, N. Blodow, and M. Beetz, "Persistent point feature histograms for 3d point clouds," in *Proceedings of the 10th International Conference on Intelligent Autonomous Systems*, Baden-Baden, Germany, 2008, pp. 119–128.

[13] A. Ramisa, G. Alenya, F. Moreno-Noguer, and C. Torras, "Finddd: A fast 3d descriptor to characterize textiles for robot manipulation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 824–830.

[14] X. Wang, X. Bai, X. Yang, W. Liu, and L. J. J. Latecki, "Maximal cliques that satisfy hard constraints with application to deformable object model learning," in *Advances in Neural Information Processing Systems*, 2011, pp. 864–872.

[15] T. Malisiewicz and A. Efros, "Beyond categories: The visual memex model for reasoning about object relationships," in *Advances in Neural Information Processing Systems*, 2009, pp. 1222–1230.

[16] S. Kumar and M. Hebert, "A hierarchical field framework for unified context-based classification," in *Proceedings of the Tenth IEEE International Conference on Computer Vision*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 1284–1291.

[17] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation," in *ECCV*, 2006, pp. 1–15.

[18] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie, "Objects in context," in *IEEE 11th International Conference on Computer Vision*, 2007, pp. 1–8.

[19] D. Munoz, J. A. D. Bagnell, N. Vandapel, and M. Hebert, "Contextual classification with functional max-margin markov networks," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2009.

[20] X. Xiong and D. Huber, "Using context to create semantic 3d models of indoor environments," in *Proceedings of the British Machine Vision Conference*, 2010, pp. 1–11.

[21] Y. Weiss, "Comparing the mean field method and belief propagation for approximate inference in mrfs," in *Saad and Opper, eds., Advanced Mean Field Methods*. MIT press, 2001.

[22] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations." San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 239–269.

[23] Y. Tsuruoka, J. Tsujii, and S. Ananiadou, "Stochastic gradient descent training for l1-regularized log-linear models with cumulative penalty," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, Stroudsburg, PA, USA, 2009, pp. 477–485.

[24] L. Bottou, "Online algorithms and stochastic approximations," in *Online Learning and Neural Networks*. Cambridge, UK: Cambridge University Press, 1998, revised, Oct. 2012.

[25] D. Munoz, N. Vandapel, and M. Hebert, "Onboard contextual classification of 3-d point clouds with learned high-order markov random fields," in *IEEE International Conference on Robotics and Automation*, May 2009, pp. 2009–2016.

[26] D. Munoz, "Inference machines: Parsing scenes via iterated predictions," Ph.D. dissertation, June 2013.

[27] S. V. N. Vishwanathan, N. N. Schraudolph, M. W. Schmidt, and K. P. Murphy, "Accelerated training of conditional random fields with stochastic gradient methods," in *International Conference on Machine Learning*, 2006, pp. 969–976.

[28] B. Dellen, F. Husain, and C. Torras, "Joint segmentation and tracking of object surfaces along human/robot manipulations," in *Int. Conf. on Comput. Vision Theory and Applicat.*, vol. 1, 2013, pp. 244–251.

[29] F. Husain, A. Colomé, B. Dellen, G. Alenyà, and C. Torras, "Realtime tracking and grasping of a moving object from range video," in *IEEE Conference on Robotics and Automation*, 2014, (To appear).

[30] A. G. Schwing, T. Hazan, M. Pollefeys, and R. Urtasun, "Distributed message passing for large scale graphical models," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1833–1840.