Nonparametric Bayesian Models for Unsupervised Scene Analysis and Reconstruction

Dominik Joho, Gian Diego Tipaldi, Nikolas Engelhard, Cyrill Stachniss, Wolfram Burgard
Department of Computer Science, University of Freiburg, Germany
{joho, tipaldi, engelhar, stachnis, burgard}@informatik.uni-freiburg.de

***Abstract*—Robots operating in domestic environments need to deal with a variety of different objects. Often, these objects are neither placed randomly, nor independently of each other. For example, objects on a breakfast table such as plates, knives, or bowls typically occur in recurrent configurations. In this paper, we propose a novel hierarchical generative model to reason about latent object constellations in a scene. The proposed model is a combination of Dirichlet processes and beta processes, which allow for a probabilistic treatment of the unknown dimensionality of the parameter space. We show how the model can be employed to address a set of different tasks in scene understanding ranging from unsupervised scene segmentation to completion of a partially specified scene. We describe how sampling in this model can be done using Markov chain Monte Carlo (MCMC) techniques and present an experimental evaluation with simulated as well as real-world data obtained with a Kinect camera.**

I. Introduction

Imagine a person laying a breakfast table and the person gets interrupted so that she cannot continue with the breakfast preparation. A service robot such as the one depicted in Fig. 1 should be able to proceed laying the table without receiving specific instructions. It faces a series of challenges: how to infer the total number of covers, how to infer which objects are missing on the table, and how should the missing parts be arranged. For this, the robot should not require any user-specific pre-programmed model but should ground its decision based on the breakfast tables it has seen in the past.

In this paper, we address the problem of scene understanding given a set of unlabeled examples and generating a plausible configuration from a partially specified scene. The *key contribution* of this paper is the definition of a novel hierarchical nonparametric Bayesian model to represent the scene structure in terms of object groups and their spatial configuration. We show how to infer the scene structure in an unsupervised fashion by using Markov chain Monte Carlo (MCMC) techniques to sample from the posterior distribution of the latent variables given the observations.

In our model, each scene contains an unknown number of latent *meta-objects*. As illustrated in Fig. 2, a meta-object is a collection of parts. In the breakfast table example, a place cover can be seen as a latent meta-object of a certain type that, for example, generates the observable objects plate, knife, and mug. A meta-object of a different type might generate a cereal bowl and a spoon. It is important to note that not all instances of the same meta-object are identical. Instances differ in the sense that some parts may be missing and that the individual parts may not be arranged in the same way.

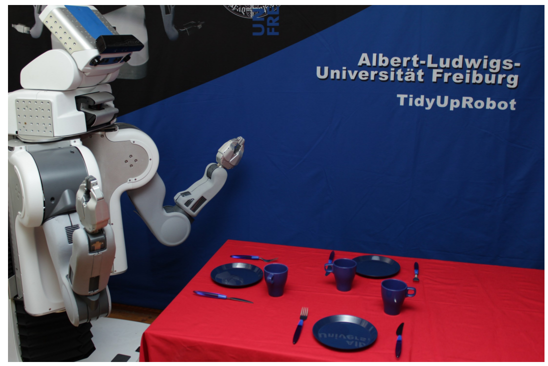


Fig. 1. A scene typically contains several observable objects and the task is to infer the latent meta-objects where a meta-object is considered to be a constellation of observable objects. At a breakfast table, for example, the meta-objects might be the covers that consist of the observable objects plate, knife, fork, and cup.

When specifying a generative model for our problem, we have the difficulty that the dimensionality of the model is part of the learning problem. This means, that besides learning the parameters of the model, like the pose of a meta-object, we additionally need to infer the *number* of involved meta-objects, parts, etc. The standard solution would be to follow the model selection approaches, for example, learning several models and then choosing the best one. Such a comparison is typically done by trading off the data likelihood with the model complexity as, for example, done for the Bayesian information criterion (BIC). The problem with this approach is the huge number of possible models, which renders this approach intractable in our case.

To avoid this complexity, we follow another approach, motivated by recent developments in the field of hierarchical nonparametric Bayesian models based on the Dirichlet process and the beta process. These models are able to adjust their dimensionality according to the given data, thereby sidestepping the need to select among several finite-dimensional model alternatives. Based on a prior over scenes, learned upon observations of training examples, the model can be used for parsing new scenes or completing partially specified scenes by sampling the missing objects.

As a motivating example, we consider learning the object constellations on a breakfast table as depicted in Fig. 1. Our model, however, is general and not restricted to this scenario.

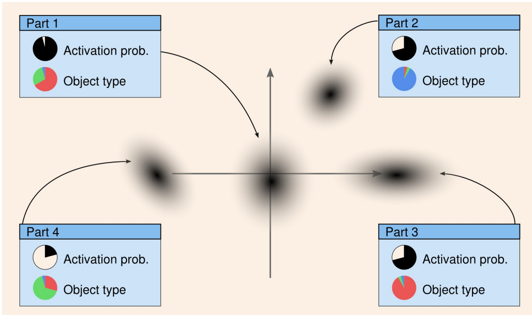


Fig. 2. A meta-object is represented as a collection of parts. Each part has Gaussian distribution, a multinomial distribution over the observable object types, and a binary activation probability.

II. Related Work

In this section we describe the relevant works on unsupervised scene analysis. A first family of approaches, and the most related to our model, employs nonparametric Bayesian models to infer spatial relations. Sudderth et al. [15] introduced the transformed Dirichlet process, a hierarchical model which shares a set of stochastically transformed clusters among groups of data. The model has been applied to improve object detection in visual scenes, given the ability of the model to reason about the number of objects and their spatial configuration. Following his paper, Austerweil and Griffiths [2] presented the transformed Indian buffet process, where they model both the features representing an object and their position relative to the object center. Moreover, the set of transformations that can be applied to a feature depends on the objects context.

A complementary approach is the use of constellation models [3, 4, 11]. Those models explicitly consider parts and structure of objects and can be learnt efficiently and in a semisupervised manner. The more successful constellation model proposed is the star model [4], a sparse representation of the object consisting of a star topology configuration of parts modeling the output of a variety of feature detectors. The main limitations of those methods, however, lies in the fact that the number of objects and parts must be defined beforehand and thus cannot be trivially used for scene understanding and object discovery. Ranganathan and Dellaert [11] used a 3D constellation model for representing indoor environments as object constellations. Another related approach to constellation models is presented in [10], where a hierarchical rule-based model is used to capture spatial relations. They also employed a star constellation model, and developed an expectation maximization (EM)-style algorithm to infer the structure and the labels of the objects and parts.

Another family of approaches relies on discriminative learning and unsupervised model selection techniques. An approach to automatic discovery of object part representation has been developed in [13]. In their work, the authors introduced a latent conditional random field (CRF) based on a flexible assembly of parts. Individual part labels are modeled as hidden nodes and a modified version of the EM algorithm has been developed for learning the pairwise structure of the underlying graphical model. Triebel et al. [19] presented a unsupervised approach to segment 3D range scan data and to discover objects by the frequency of their parts appearance. The data is first segmented using graph-based clustering, then each segment is treated as a potential object part. The authors then used CRFs to represent both the part graph to model the interdependence of parts with object labels, and a scene graph to smooth the object labels. Spinello et al. [14] proposed an unsupervised detection technique based on a voting scheme of image descriptors. They introduced the concept of *latticelets*: a minimal set of pairwise relations that can generalize the patterns connectivity. Conditional random fields are then used to couple low level detections with high level scene description. Jiang et al. [8] used an undirected graphical model to infer the best placement for multiple objects in a scene. Their model considers several features of object configurations, such as stability, stacking, and semantic preferences.

A different approach is the one of Fidler and Leonardis [5]. They proposed an approach to constructing a hierarchical representation of visual input. Their approach uses a bottom up strategy, where they learn the statistically most significant compositions of simple features with respect to more complex higher level ones. Parts are learned sequentially, layer after layer. Separate classification and grouping technique are used for the bottom and top layers to account for the numerical difference (sensor data) and semantical ones (object category).

However, to the best of the our knowledge, no previous model is able to deal with a nested hierarchy of different objects and group types and being flexible enough to allow for both scene understanding and generation.

III. Generative Scene Model

In this section, we describe the proposed generative scene model. We assume that the reader is familiar with the basics of Bayesian nonparametric models [6], especially with the Chinese restaurant process (CRP) and the Dirichlet process (DP) [16], the (two-parameter) Indian buffet process (IBP) and the beta process (BP) [7, 18], and the concepts of hierarchical [17] and nested [12] processes in this context.

In the following, we consider a scene as a collection of observable objects represented as labeled points in the 2D plane. The 2D asssumption is due to our motivation to model table scenes. However, the model is not specifically geared towards 2D data and could in principle also be applied to 3D data. We assume that each scene contains an unknown number of latent meta-objects, which is a collection of parts, and that we need to infer the number of latent meta-objects when learning the model. The observable objects of a scene can be grouped into clusters and each cluster corresponds to an *instance of a meta-object* that generated the observable objects of its cluster. Each part generates at most one observable object. A part consists of a Gaussian distribution that constrains the relative position of an observable object with respect to a reference frame that is defined by the meta-object. Further, each part is associated with a multinomial distribution

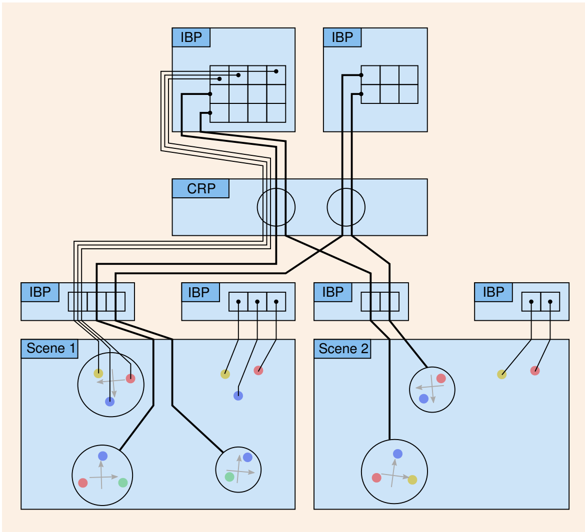


Fig. 3. Basic structure of our model: The Indian buffet processes (IBP) on top are nested within the tables (represented as circles) of the Chinese restaurant process (CRP) below them. The blocks within the IBP frames represent the relevant part of the IBP matrix (see Fig. 4). Clusters in the scenes are meta-objects and objects (colored points) of the clusters need to be associated to entries in the IBPs shown on top. This is explicitly shown for one cluster in the first scene. For visibility reasons, the rest of the associations are drawn as a single thick line. At the lowest level there are the IBPs that draw the meta-objects and noise objects for each scene.

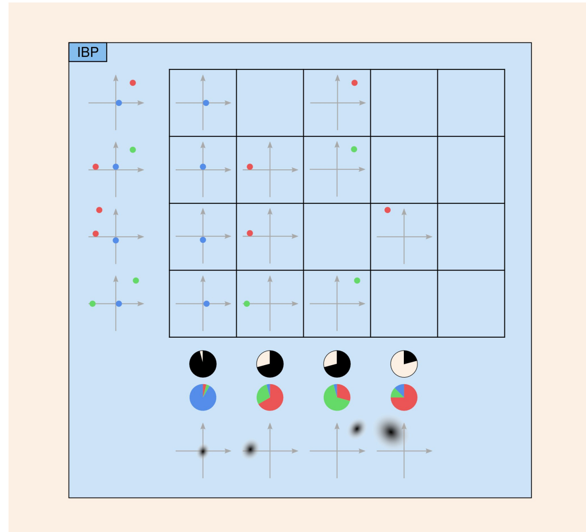


Fig. 4. A more detailed view on an IBP representing a meta-object type. This would correspond to one of the IBPs at the very top in Fig. 3. The rows represent customers and the columns represent dishes (parts). The customers of this IBP are the meta-object *instances* associated to this meta-object *type* in any of the scenes. The objects of the instances must be associated to one of the parts. They thereby update the posterior predictive distribution (shown at the bottom) over the objects of a new customer. Remember, that the actual part parameters are integrated out due to the usage of conjugate priors.

over the type of the observable objects to be generated at this relative location and, lastly, a part is associated with a binary activation probability that specifies how likely it is that this part generates the object when the meta-object is being instantiated in a scene. A prototypical meta-object is illustrated in Fig. 2. Please note, that we will use the terms "meta-object" and "cover" interchangeably.

A. Description of the Generative Process

Following the example given in the introduction, imagine that our robot's goal is to set a table for a typical family breakfast. At the beginning, it enters a room with an empty breakfast table and an infinite number of side tables, each holding a prototypical cover. It estimates the area A of the breakfast table surface and boldly decides that $n \sim \mathrm{Poisson}(A\lambda)$ covers are just right. The robot chooses one of the side tables and finds a note with the address of a Chinese restaurant where it can get the cover. Arriving there, it sees again infinitely many tables each corresponding to a particular cover type and each displaying a count of how often someone took a cover from this table. It is fine with just about any cover type and decides randomly based on the counts displayed at the tables, even considering a previously unvisited table. At that table there is another note redirecting to an Indian restaurant. In this restaurant, it is being told that the cover needs to be assembled by choosing the parts that make up this cover. The robot randomly selects the parts based on their popularity and even considers to use a few parts no one has ever used before. For each chosen part its relative position is sampled from the part's Gaussian distribution and then the robot samples the object from the part's multinomial distribution over types (plate, knife, fork, etc.). Having assembled the cover this way, the robot returns to the breakfast table and puts it randomly on it. The process is then repeated for the remaining $n-1$ covers. See Fig. 3 for an overview of the model structure.

More formally, we have a hierarchical model with a high-level Dirichlet process DP_t, a low-level beta process BP_c and a further independent beta process BP_ϵ. First, we draw G_t from the high-level DP_t

$$G_t \sim \mathrm{DP}_t(\alpha_t, \mathrm{BP}_p(c_p, \alpha_p, \mathrm{Dir} \times \mathcal{NW})), \tag{1}$$

where the base distribution of DP_t is a beta process BP_p modeling the parts' parameters. This is done only once and all scenes to be generated will make use of the same draw G_t. This draw describes the distribution over all possible cover types and corresponds to the Chinese restaurant mentioned above. The base distribution of the beta process is the prior distribution over the part parameters. The parameters are a 2D Gaussian distribution over the relative location and a multinomial over the observable object types. The parameters are sampled independently and we use their conjugate priors in the base distribution, i.e., a (symmetric) Dirichlet distribution Dir for the multinomial and the normal-Wishart distribution [9] $\mathcal{NW}$ for the 2D Gaussian. The mass parameter α_p of BP_p is our prior over the number of activated parts of a single cover instance and the concentration parameter c_p influences the total number of instantiated parts across all instances of the same type. Likewise, the parameter α_t influences the expected number of cover types. Each scene s has its own meta-object (cover) IBP and the cover instances are determined by a single draw from the corresponding beta-Bernoulli process as follows:

$$G_c^{(s)} \sim \mathrm{BP}_c(1, |A_s|\,\alpha_c,\, G_t \times U(A_s \times [-\pi, \pi])) \tag{2}$$

$$\{G_{t_j}, T_j\}_j \sim \mathrm{BeP}(G_c^{(s)}) \tag{3}$$

$$\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \boldsymbol{\gamma}_k\}_k \sim \mathrm{BeP}(G_{t_j}) \qquad \text{for each } j \tag{4}$$

$$\{\mathbf{x}, \omega\} \sim p(\mathbf{z} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \boldsymbol{\gamma}_k, T_j) \qquad \text{for each } k \tag{5}$$

In Eq. (2), the concentration parameter is irrelevant and arbitrarily set to one. The mass parameter $|A_s|\,\alpha_c$ is the expected number of cover instances in a scene. The base distribution of BP_c samples the parameters of a cover j: its type t_j and its pose T_j. The type t_j is drawn from the distribution over cover types G_t from Eq. (1). The pose T_j is drawn from a uniform distribution U over the pose space $A_s \times [-\pi, \pi]$ where A_s is the table area. Each atom selected by the Bernoulli process in Eq. (3) corresponds to a cover instance and this selection process corresponds to the side-table metaphor mentioned above. The cover type t_j basically references a draw G_{t_j} from the nested beta process in Eq. (1) which models the parts of a cover type. Thus, in Eq. (4) we need another draw from a Bernoulli process to sample the activated parts for this cover instance, which yields the Gaussians $(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ and the multinomials $(\boldsymbol{\gamma}_k)$ for each active part k. Finally, in Eq. (5) we draw the actual observable data from the data distribution as realizations from the multinomials and the (transformed) Gaussians, each yielding an object $\mathbf{z} = \{\mathbf{x}, \omega\}$ on the table with location $\mathbf{x}$ and type ω.

Each scene has an additional independent beta process BP_ϵ

$$G_\epsilon^{(s)} \sim \mathrm{BP}_\epsilon(1, \alpha_\epsilon, M \times U(A_s)) \tag{6}$$

$$\{\mathbf{x}_i, \omega_i\}_i \sim \mathrm{BeP}(G_\epsilon^{(s)}) \tag{7}$$

that directly samples objects (instead of covers / meta-objects) at random locations in the scene. Here, M is a multinomial over the observable object types and $U[A_s]$ is the uniform distribution over the table area A_s. This beta-Bernoulli process will mainly serve as a "noise model" during MCMC inference to account for yet unexplained objects in the scene. Accordingly, we set the parameter α_ϵ to a rather low value to penalize scenes with many unexplained objects. Figs. 3 and 4 illustrate the overall structure of the model.

B. Posterior Inference in the Model

In this section, we describe how to sample from the posterior distribution over the latent variables $\{\mathbf{C}, \mathbf{a}\}$ given the observations $\mathbf{z}$. We use the following notation. The observations are the objects of all scenes and a single object $\mathbf{z}_i = \{\mathbf{x}_i, \omega_i\}$ has a 2D location $\mathbf{x}_i$ on the table and a discrete object type ω_i. A meta-object instance j has parameters $C_j = \{T_j, t_j, \mathbf{d}_j\}$, where T_j denotes the pose and t_j denotes the meta-object type, which is an index to a table in the type CRP, and $\mathbf{d}_j$ denotes part activations and associations to the observable objects. If $d_{j,k} = 0$ then part k of meta-object instance j is inactive, where k is an index to a dish in the corresponding part IBP (which is nested in the CRP table t_j). Otherwise, the part is active and $d_{j,k} \neq 0$ is a reference to the associated observable object, i.e., $d_{j,k} = i$ if it generated the observation $\mathbf{z}_i$. Next, for each scene we have the associations $\mathbf{a}$ to the noise process, i.e., the list of objects currently not associated to any meta-object instance. For ease of notation, we will use index functions $[\cdot]$ in an intuitive way, for example, $\mathbf{t}_{[-j]}$ denotes all type assignments except the type assignment t_j of meta-object j, and $\mathbf{T}_{[t_j]}$ denotes all poses of meta-objects that have the same type t_j as meta-object j, and $\mathbf{z}_{[\mathbf{d}]}$ are all observations referenced by the associations $\mathbf{d}$, etc.

We employ Metropolis-Hastings (MH) steps to sample from the posterior [1], which allows for big steps in the state space by updating several strongly correlated variables at a time. In the starting state of the Markov chain, all objects are assigned to the noise IBP of their respective scene and thus are interpreted as yet unexplained objects. The sampler is run for a fixed yet sufficiently high number of iterations to be sure that the Markov chain converged. We use several different types of MH moves, which we will explain in the following after describing the joint probability.

Joint Probability: $p(\mathbf{C}, \mathbf{a}, \mathbf{z})$. The joint probability is

$$p(\mathbf{C}, \mathbf{a}, \mathbf{z}) = \left(\prod_{s=1}^{S} p(n_{s,\epsilon}) p(n_{s,m})\right) p(\mathbf{t}) \left(\prod_{t=1}^{n_t} p(\mathbf{d}_{[t]} \mid \mathbf{t})\right)$$
$$p(\mathbf{T}) \left(\prod_{t=1}^{n_t} \prod_{k=1}^{K_t} p(\mathbf{z}_{[\mathbf{d}_{[t,k]}]} \mid \mathbf{T}_{[t]}, \mathbf{d}_{[t,k]}, \mathbf{t})\right). \tag{8}$$

Here, $n_{s,m}$ is the number of meta-object instances and $n_{s,\epsilon}$ is the number of noise objects in scene s. Each dish parameter of the noise IBP directly corresponds to the parameters $\mathbf{z}_i = \{\mathbf{x}_i, \omega_i\}$ of the associated noise object, as we assumed that there is no data distribution associated with these dishes. The base distribution for the dish parameters consists of independent and uniform priors over the table area and the object types, and so each dish parameter has probability $(n_\omega\,|A_s|)^{-1}$, where n_ω is the number of observable object types and $|A_s|$ is the area of the table in scene s. Thus, the probability of the objects $\mathbf{z}_{[\mathbf{a}_{[s]}]}$ associated to a scene's noise IBP only depends on the *number* of noise objects and not on the particular type or position of these objects. However, it would be straightforward to use a non-uniform base distribution. Denoting the Poisson distribution with mean λ as $\mathrm{Poi}(\cdot \mid \lambda)$ we thus have

$$p(n_{s,\epsilon}) = p(\mathbf{z}_{[\mathbf{a}_{[s]}]}, \mathbf{a}_{[s]}) = n_{s,\epsilon}!\, \mathrm{Poi}(n_{s,\epsilon} \mid \alpha_\epsilon)(n_\omega\,|A_s|)^{-n_{s,\epsilon}}. \tag{9}$$

Next, $p(n_{s,m}) = n_{s,m}!\,\mathrm{Poi}(n_{s,m} \mid \alpha_m\,|A_s|)$ is the prior probability for having $n_{s,m}$ meta-objects in scene s. The dish parameters of a meta-object IBP are the meta-object parameters C_j, consisting of the pose, type, and part activations. The probability for sampling a pose $p(T_j) = (|A_s|\,2\pi)^{-1}$ is uniform over the table surface and uniform in orientation, hence $p(\mathbf{T}) = \prod_{s=1}^{S}(|A_s|\,2\pi)^{-n_{s,m}}$. Next, $p(\mathbf{t})$ is the CRP prior for the meta-object types of all meta-object instances. The factors $p(\mathbf{d}_{[t]} \mid \mathbf{t})$ are the IBP priors for the part activations for all meta-object instances of a certain type t and there are n_t different types instantiated in total. During MCMC inference, we will exclusively need the conditional distribution for a single meta-object j

$$p(t_j, \mathbf{d}_j \mid \mathbf{t}_{[-j]}, \mathbf{d}_{[-j]}) = p(\mathbf{d}_j \mid \mathbf{d}_{[-j,t_j]}, \mathbf{t}) p(t_j \mid \mathbf{t}_{[-j]}). \quad (10)$$

Here, $p(t_j \mid \mathbf{t}_{[-j]})$ is the CRP predictive distribution

$$p(t_j = i \mid \mathbf{t}_{[-j]}) = \begin{cases} \frac{n_i}{\alpha_c + \sum_{i'} n_{i'}} & t_j \text{ is an existing type} \\ \frac{\alpha_c}{\alpha_c + \sum_{i'} n_{i'}} & t_j \text{ is a new type} \end{cases}, \quad (11)$$

and n_i is the number of meta-object instances of type i (not counting instance j) and α_c is the concentration parameter of the CRP. Further, $p(\mathbf{d}_j \mid \mathbf{d}_{[-j,t_j]}, \mathbf{t})$ is the predictive distribution of the two-parameter IBP, which factors into activation probabilities for each of the existing parts and an additional factor for the number of new parts. An existing part is a part that has been activated by at least one other meta-object instance of this type in any of the scenes. The activation probability for an existing part k is

$$p(d_{j,k} \neq 0 \mid \mathbf{d}_{[-j,t_j]}, \mathbf{t}) = \frac{n_k}{n_{t_j} + c_p}, \quad (12)$$

where c_p is the concentration parameter of the two-parameter IBP, n_k is the number of meta-object instances that have part k activated in any of the scenes, and n_{t_j} is the total number of meta-object instances of type t_j in all scenes (the counts exclude the meta-object j itself). The probability for having n_{K^+} associations to new parts is

$$p(\mathbf{d}_{[j,K^+]} \mid \mathbf{d}_{[-j,t_j]}, \mathbf{t}) = n_{K^+}! \,\mathrm{Poi}\left(n_{K^+} \middle| \frac{c_p \alpha_p}{n_{t_j} + c_p}\right), \quad (13)$$

where α_p is the mass parameter of the two-parameter IBP, and $\mathbf{d}_{[j,K^+]}$ denotes the associations to new parts.

As stated in Eq. (8), the data likelihood $p(\mathbf{z}_{[\mathbf{d}]} \mid \mathbf{T}, \mathbf{d}, \mathbf{t})$ for the objects associated to meta-objects factors into likelihoods for each individual part k. Further, it factors into a spatial component and a component for the observable object type. As the meta-object poses $\mathbf{T}$ are given, we can transform the absolute positions $\mathbf{x}_{[\mathbf{d}_{[t,k]}]}$ of the objects associated to a certain part k of meta-object type t into relative positions $\tilde{\mathbf{x}}_{[\mathbf{d}_{[t,k]}]}$ with respect to a common meta-object reference frame. The relative positions are assumed to be sampled from the part's Gaussian distribution which in turn is sampled from a normal-Wishart distribution. As the Gaussian and the normal-Wishart distribution form a conjugate pair, we can analytically integrate out the part's Gaussian distribution which therefore does not have to be explicitly represented. Hence, the joint likelihood for $\tilde{\mathbf{x}}_{[\mathbf{d}_{[t,k]}]}$ of part k is computed as the marginal likelihood under a normal-Wishart prior. During MCMC inference, we only need to work with the posterior predictive distribution

$$p(\tilde{\mathbf{x}}_{[d_{j,k}]} \mid \tilde{\mathbf{x}}_{[\mathbf{d}_{[-j,t_j,k]}]}) = t_\nu(\tilde{\mathbf{x}}_{[d_{j,k}]} \mid \mu, \Sigma) \quad (14)$$

for a single relative position given the rest. This is a multivariate t-distribution t_ν with parameters μ, Σ depending both on $\tilde{\mathbf{x}}_{[\mathbf{d}_{[-j,t_j,k]}]}$ and the parameters of the normal-Wishart prior – for details see [9]. Similarly, the part's multinomial distribution over the observable object types can be integrated out as it forms a conjugate pair with the Dirichlet distribution. The posterior predictive distribution for a single object type is

$$p(\omega_{[d_{j,k}]} \mid \omega_{[\mathbf{d}_{[-j,t_j,k]}]}) = \frac{n_\omega + \alpha_\omega}{\sum_{\omega'}(n_{\omega'} + \alpha_{\omega'})}, \quad (15)$$

where n_ω is the number of times an object of type ω has been associated to part k of this meta-object type t_j, and α_ω is the pseudo-count of the Dirichlet prior. When describing the MCMC moves, we will sometimes make use of the predictive likelihood for all objects $\mathbf{z}_{[\mathbf{d}_j]}$ associated to a single meta-object instance j. This likelihood factors into the posterior predictive distributions of the individual parts and their spatial and object type components, as described above. In the following, we will describe the various MCMC moves in detail.

Death (birth) move: $(T_j, t_j, \mathbf{d}_j, \mathbf{a}) \to (\mathbf{a}^\star)$. A death move selects a meta-object j uniformly at random, adds all of its currently associated objects to the noise process and removes the meta-object j from the model. The proposal probability for this move is $q_d(\mathbf{C}_{-j}, \mathbf{a}^\star \mid C_j, \mathbf{C}_{-j}, \mathbf{a}) = (n_m)^{-1}$ where n_m denotes the number of instantiated meta-objects in this scene before the death move. To simplify notation, we will just write $q_d(C_j)$ for the probability of deleting meta-object j. The reverse proposal is the birth proposal $q_b(C_j^\star, \mathbf{C}_{-j}, \mathbf{a}^\star \mid \mathbf{C}_{-j}, \mathbf{a}, \mathbf{z})$ that proposes new parameters $C_j^\star = \{T_j^\star, t_j^\star, \mathbf{d}_j^\star\}$ for an additional meta-object: the pose $T_j^\star$, the type $t_j^\star$, and the associations $\mathbf{d}_j^\star$. The new meta-object may reference any of the objects previously associated to the noise process and any non-referenced noise objects remain associated to the noise process. We will describe the details of the birth proposal in detail later on. To simplify notation, we will just write $q_b(C_j)$ for the birth proposal. Plugging in the model and proposal distributions in the MH ratio and simplifying we arrive at the acceptance ratio of the death move

$$R_d = \frac{1}{p(\mathbf{z}_{[\mathbf{d}_j]} \mid \mathbf{z}_{[\mathbf{d}_{[-j,t_j]}]}, \mathbf{T}_{[t_j]}, \mathbf{d}_{[t_j]}, \mathbf{t}) p(t_j, \mathbf{d}_j \mid \mathbf{t}_{[-j]}, \mathbf{d}_{[-j]})} \frac{1}{p(T_j)} \frac{p(n_m - 1)}{p(n_m)} \frac{p(n_\epsilon + n_j)}{p(n_\epsilon)} \frac{q_b(C_j)}{q_d(C_j)}. \quad (16)$$

The counts n_j, n_m, and n_ϵ refer to the state before the death move, and n_m denotes the number of meta-objects in this scene, n_j are the number of objects currently associated to meta-object j, and n_ϵ is the number of noise objects. The ratio of a birth move is derived similarly.

Switch move: $(T_j, t_j, \mathbf{d}_j, \mathbf{a}) \to (T_j^\star, t_j^\star, \mathbf{d}_j^\star, \mathbf{a}^\star)$. This move is a combined death and birth move. It removes a meta-object and then proposes a new meta-object using the birth proposal. Thus, the number of meta-objects remains the same but one meta-object simultaneously changes its type t_j, pose T_j, and part associations $\mathbf{d}_j$. The death proposals cancel out and the acceptance ratio of this move is

$$R_s = \frac{p(\mathbf{z}_{[\mathbf{d}_j^\star]} \mid \mathbf{z}_{[\mathbf{d}_{[-j,t_j^\star]}]}, T_j^\star, \mathbf{T}_{[-j,t_j^\star]}, \mathbf{d}_j^\star, \mathbf{d}_{[-j,t_j^\star]}, t_j^\star, \mathbf{t}_{[-j]})}{p(\mathbf{z}_{[\mathbf{d}_j]} \mid \mathbf{z}_{[\mathbf{d}_{[-j,t_j]}]}, T_j, \mathbf{T}_{[-j,t_j]}, \mathbf{d}_j, \mathbf{d}_{[-j,t_j]}, t_j, \mathbf{t}_{[-j]})} \frac{p(t_j^\star, \mathbf{d}_j^\star \mid \mathbf{t}_{[-j]}, \mathbf{d}_{[-j]}) p(T_j^\star) p(n_\epsilon^\star) q_b(C_j)}{p(t_j, \mathbf{d}_j \mid \mathbf{t}_{[-j]}, \mathbf{d}_{[-j]}) p(T_j) p(n_\epsilon) q_b(C_j^\star)}. \quad (17)$$

Shift move: $(T_j) \rightarrow (T_j^\star)$. This move disturbs the pose T_j of a meta-object by adding Gaussian noise to it while the type and part associations remain unchanged. The acceptance ratio depends only on the spatial posterior predictive distributions of the objects associated to this meta-object. The proposal probabilities cancel due to symmetry and the final ratio is

$$R_T = \frac{p(\mathbf{x}_{[\mathbf{d}_j]} \mid \mathbf{x}_{[\mathbf{d}_{[-j,t_j]}]}, T_j^\star, \mathbf{T}_{[-j,t_j]}, \mathbf{d}_{[t_j]}, \mathbf{t}) p(T_j^\star)}{p(\mathbf{x}_{[\mathbf{d}_j]} \mid \mathbf{x}_{[\mathbf{d}_{[-j,t_j]}]}, T_j, \mathbf{T}_{[-j,t_j]}, \mathbf{d}_{[t_j]}, \mathbf{t}) p(T_j)}. \quad (18)$$

Association move: $(\mathbf{d}_j, \mathbf{a}) \rightarrow (\mathbf{d}_j^\star, \mathbf{a}^\star)$. This move samples the individual part associations of a single meta-object j. In the IBP metaphor, this corresponds to sampling the selected dishes (activated parts) of this meta-object. We need to split this sampling into two stages: first, we use Gibbs sampling to obtain the association for each existing part and, second, we use MH moves for sampling associations to new parts.

In the first stage, we proceed for each existing part as follows. If a part is already associated with an object, we consider this object to be temporarily re-associated to the noise process (such that there are now n_ϵ noise objects). We then use Gibbs sampling to obtain one of $n_\epsilon + 1$ possible associations: either the part k is inactive ($d_{j,k} = 0$) and not associated with any object, or it is active ($d_{j,k} \neq 0$) and associated with one out of n_ϵ currently available noise objects (e.g. $\mathbf{z}_i$ when $d_{j,k} = i$). The probabilities for these cases are proportional to

$$p(d_{j,k} = i) \propto \begin{cases} p(d_{j,k} = 0 \mid \mathbf{d}_{[-j,t_j,k]}, \mathbf{t}) p(n_\epsilon) & i = 0 \\ p(\mathbf{z}_i \mid \mathbf{z}_{[\mathbf{d}_{[-j,t_j,k]}]}, \mathbf{T}_{[t_j]}, d_{j,k}, \mathbf{d}_{[-j,t_j,k]}, \mathbf{t}) & \\ p(d_{j,k} \neq 0 \mid \mathbf{d}_{[-j,t_j,k]}, \mathbf{t}) p(n_\epsilon - 1) & i \neq 0 \end{cases} \quad (19)$$

In the second stage, we resample the associations to *new* parts. For this, we use two complementary MH moves: one move increases the number of new parts by one by assigning a noise object to a new part, while the other move decreases the number of new parts by one by assigning an associated object (of a new part) to the noise process. The acceptance ratio for removing object $\mathbf{z}_i$ from a new part k of meta-object j that currently has n_{K+} new parts is

$$R_- = \frac{1}{p(\mathbf{z}_i \mid T_j, \mathbf{d}_{[j,K+]})} \frac{p(\mathbf{d}^\star_{[j,K+]} \mid \mathbf{d}_{[-j,t_j]}, \mathbf{t})}{p(\mathbf{d}_{[j,K+]} \mid \mathbf{d}_{[-j,t_j]}, \mathbf{t})} \frac{p(n_\epsilon + 1)}{p(n_\epsilon)} \frac{q_+}{q_-} \quad (20)$$

The proposal $q_- = (n_{K+})^{-1}$ chooses one of the new parts to be removed uniformly at random, while the reverse proposal $q_+ = (n_\epsilon + 1)^{-1}$ chooses uniformly at random one of then $n_\epsilon + 1$ noise objects to be associated to a new part. The MH move that increases the number new parts is derived similarly.

Birth proposal. $q_b(T_j, t_j, \mathbf{d}_j)$ The birth, death, and switch moves rely on a birth proposal that samples new meta-object parameters $C_j = \{T_j, t_j, \mathbf{d}_j\}$. The general idea is to sample the pose T_j and type t_j in a first step. We then proceed to sample the associations $\mathbf{d}_j$ given T_j and t_j, i.e., the potential assignment of noise objects to the parts of this meta-object.

Sampling T_j and t_j is done in either of two modes: the *object mode* or the *matching mode*. In object mode, we choose an object uniformly at random and center the meta-object pose T_j at the object's location (with random orientation) and add Gaussian noise to it. The type t_j is sampled from the current predictive distribution of the type CRP. In contrast, the matching mode was inspired by bottom-up top-down approaches and aims to propose T_j and t_j in a more efficient way by considering the currently instantiated meta-object types in the model. However, in contrast to the object mode, it cannot propose new meta-object types (new tables in the type CRP). It selects two objects and associates them to a suitable part pair. This suffices to define the pose T_j as the corresponding transformation of these parts into the scene. In detail, we first sample one of the objects $\mathbf{z}_i$ at random and choose its nearest neighbor $\mathbf{z}_j$. We match this ordered pair $\langle \mathbf{z}_i, \mathbf{z}_j \rangle$ against all ordered part pairs $\langle k_{i'}, k_{j'} \rangle$ of the meta-object types to obtain the matching probabilities with respect to the parts' posterior predictive means, $\boldsymbol{\mu}_{i'}$ and $\boldsymbol{\mu}_{j'}$, of the spatial distribution, and their posterior predictive distributions, $M_{i'}$ and $M_{j'}$, over the observable object types ω_i and ω_j

$$p_m(\langle \mathbf{z}_i, \mathbf{z}_j \rangle, \langle k_{i'}, k_{j'} \rangle) = \mathcal{N}(d_\triangle \,|\, 0, \sigma_m^2) M_{i'}(\omega_i) M_{j'}(\omega_j). \quad (21)$$

Here, $d_\triangle = \|\mathbf{x}_i - \mathbf{x}_j\| - \|\boldsymbol{\mu}_{i'} - \boldsymbol{\mu}_{j'}\|$ is the residual of the objects' relative distance w.r.t. the distance of the posterior means and σ_m^2 is a fixed constant. We then sample a part pair $\langle k_{i'}, k_{j'} \rangle$ proportionally to its matching probability to define, together with the objects $\mathbf{z}_i$ and $\mathbf{z}_j$, the pose T_j. Finally, we add Gaussian noise to T_j. The sampled part pair implicitly defines the meta-object type t_j.

After having sampled T_j and t_j using either of these two modes, the next step is to sample the associations $\mathbf{d}_j$. For this, we randomly choose, without repetition, a noise object and use Gibbs sampling to obtain its association to either: (a) a yet unassociated part of the meta-object instance; (b) a new part of the meta-object instance; or (c) be considered a noise object. The probabilities for (a) and (b) are proportional to the spatial and object type posterior predictive distributions of the respective parts, while (c) is based on the noise IBP's base distribution.

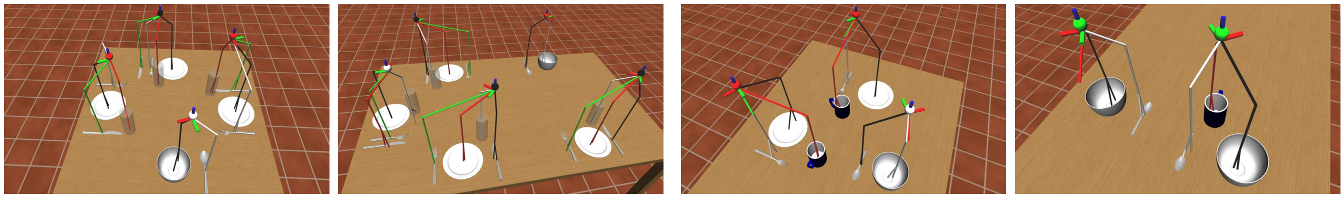
Besides sampling from the proposal distribution we also need to be able to evaluate the likelihood $q_b(C_j)$ for sampling a given parameter set C_j. For this, we need to marginalize over the latent variables of the proposal, e.g. the binary mode variable (*object mode* or *matching mode*), the chosen object $\mathbf{z}_i$, and the chosen part pair of the *matching mode*.

IV. EXPERIMENTS

We tested our model on both synthetic data and real-world data acquired with a Kinect camera.

A. Synthetic Data

In the synthetic data experiment, table scenes were generated automatically using a different, hand-crafted generative model. We generated 25 training scenes ranging from two to six covers. The cover types represent two different breakfast types. The first type is composed by a cereal bowl and a spoon, while the second type is composed by a plate with a glass, a



(a) Examples of parsed scenes of the synthetic data set.

(b) Examples of parsed scenes of the real-world data set.

Fig. 5. The picture shows the inferred meta-objects with their part relationships. Each picture shows a different scene from a different MCMC run.

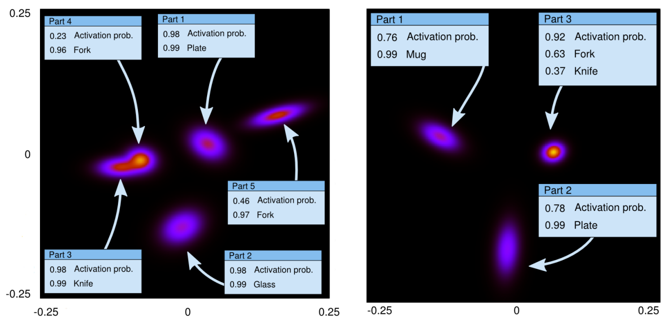


Fig. 6. Spatial posterior predictive distributions of several parts of a cover type and their activation probabilities and most likely observable object types. The cover type on the left was learned on synthetic data, while the one on the right was learned on the real-world data set.

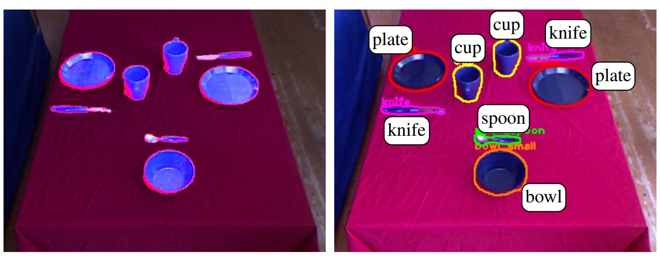
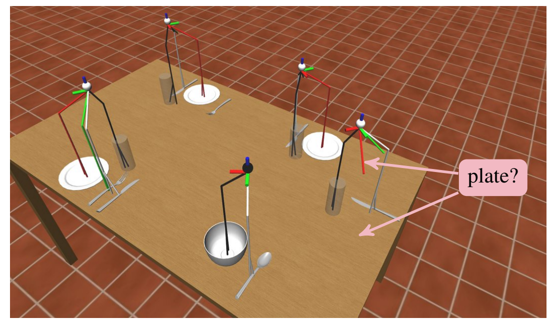


Fig. 7. Segmentation (left) and classification (right) results on the point cloud segments visualized in the Kinect image.

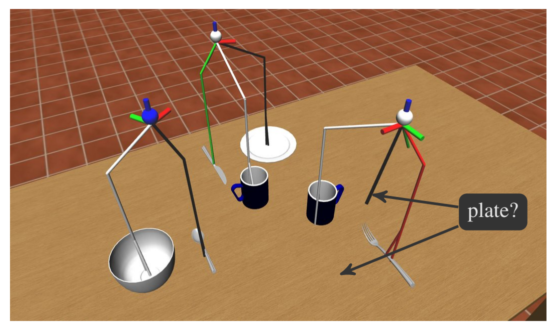
fork and a knife. The fork can be randomly placed at the right or the left of the plates or being absent.

The latent parameters are inferred using all the generated training scenes and setting the hyperparameters to $\alpha_\epsilon = 0.5$, $c_p = 0.25$,$\alpha_p = 2.5$, $\alpha_t = 5$, and $\alpha_c = 1$. As a first test, we wanted to show that the model is able to segment the scenes in a consistent and meaningful way. The results of this test are shown in Fig. 5a. A set of different scenes are segmented by using the learned model. We see that each scene has been segmented with the same cover types and objects are correctly clustered. Note that the color of the meta-object and of the part can change in every run, since the ordering of types and meta-objects is not relevant in our model. What is important is that the topological and metrical configuration are respected.

A second test is to see if the model can infer the meta-objects in incomplete scenes and if it is able to complete them. To this end, we artificially eliminated objects from already generated scenes and segmented the altered scene again using



(a) Incomplete scene of the synthetic data set.



(b) Incomplete scene of the real data set.

Fig. 8. Inferred meta-objects in an incomplete scene of the synthetic data set (a) and the real data set (b). The color of the spheres indicate the meta-object type and the color of the lines indicate the part type. In both scenes one plate has been removed. The model is still able to correctly segment the incomplete scenes and infer the missing objects. To enhance readability the pictures have been modified with human readable labels.

the learned model. We discovered that the approach was able to infer the correct cover type even in the absence of the missing object. As can be seen in Fig. 8a, the approach correctly segmented and recognized the meta-objects and is aware of the missing plate (the red branch of the hierarchy that is not grounded on an existing object). The missing object can then be inferred by sampling from the part's posterior predictive distribution over the location and type of the missing object, as the one shown in Fig. 6 (left).

B. Real-world Data

We used a Microsoft Kinect depth camera to identify the objects on the table by, first, segmenting the objects by subtracting the table plane in combination with a color-based segmentation. Second, we detect the objects based on the segmented pointcloud using a straight forward feature-based

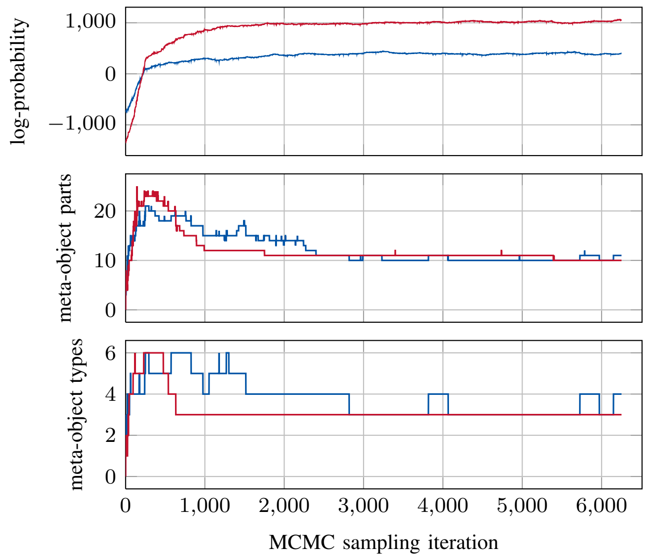


Fig. 9. The plots depict the log-probability, the number of meta-object parts (summed over all meta-object types), and the number of meta-object types as they evolve during MCMC sampling. The red lines corresponds to the synthetic data set and the blue lines to the real-world data set.

object detection system with a cascade of one-vs-all classifiers. An example for the segmentation and object identification is shown in Fig. 7. Note that there are likely to be better detection systems, however, the task of detecting the objects on the table is orthogonal to the scientific contribution of this paper.

The same set of tests performed on the synthetic data have been performed also in this case, showing basically the same results. In particular, Fig. 5b shows the segmentation results, Fig. 8b shows a modified scene where a plate has been removed and Fig. 6 (right) shows the posterior predictive distribution for location and type, as for the synthetic case, that can be used to complete an incomplete scene.

To better illustrate the inference process, we plot the log-probability and the number of parts, and number of meta-object types in Fig. 9 as they evolve during MCMC sampling.

V. Conclusion

This paper presents a novel and fully probabilistic generative model for unsupervised scene analysis. Our approach is able to model the spatial arrangement of objects. It maintains a nonparametric prior over scenes, which is updated by observing scenes, and can directly be applied for model completion by inferring missing objects in an incomplete scene. Our model applies a combination of Dirichlet processes and beta processes, allowing for a probabilistic treatment of the dimensionality of the model. In this way, we avoid a model selection step, which is typically intractable for the models considered here. To evaluate our approach, we successfully applied it to the problem of adding missing objects in the context of the task of laying a table.

Acknowledgments

This work has been supported by the Deutsche Forschungsgemeinschaft (DFG) under contract number SFB/TR 8 Spatial Cognition (R6-[SpaceGuide]) and by the EC under contract number FP7-ICT-248258-First-MM.

References

[1] C. Andrieu, N. de Freitas, A. Doucet, and M. I. Jordan. An Introduction to MCMC for Machine Learning. *Machine Learning*, 50(1–2):5–43, 2003.

[2] J. L. Austerweil and T. L. Griffiths. Learning invariant features using the Transformed Indian Buffet Process. In *Advances in Neural Information Processing Systems (NIPS)*, 2010.

[3] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2003.

[4] R. Fergus, P. Perona, and A. Zisserman. A Sparse Object Category Model for Efficient Learning and Exhaustive Recognition. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[5] S. Fidler and A. Leonardis. Towards Scalable Representations of Object Categories: Learning a Hierarchy of Parts. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[6] S. J. Gershman and D. M. Blei. A tutorial on Bayesian nonparametric models. *Journal of Mathematical Psychology*, 56(1):1–12, 2012.

[7] T. L. Griffiths and Z. Ghahramani. Infinite latent feature models and the Indian buffet process. In *Advances in Neural Information Processing Systems (NIPS)*. 2006.

[8] Y. Jiang, M. Lim, C. Zheng, and A. Saxena. Learning to place new objects in a scene. *Int. Journal of Robotics Research (IJRR)*, 2012. To appear.

[9] K. P. Murphy. Conjugate Bayesian analysis of the univariate Gaussian: a tutorial. Self-published notes, 2007.

[10] D. Parikh, C. L. Zitnick, and T. Chen. Unsupervised Learning of Hierarchical Spatial Structures In Images. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[11] A. Ranganathan and F. Dellaert. Semantic Modeling of Places using Objects. In *Proc. of Robotics: Science and Systems (RSS)*, Atlanta, GA, USA, 2007.

[12] A. Rodríguez, D. B. Dunson, and A. E. Gelfand. The Nested Dirichlet Process. *Journal of the American Statistical Association*, 103(483):1131–1154, 2008.

[13] P. Schnitzspan, S. Roth, and B. Schiele. Automatic Discovery of Meaningful Object Parts with Latent CRFs. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[14] L. Spinello, R. Triebel, D. Vasquez, K. O. Arras, and R. Siegwart. Exploiting Repetitive Object Patterns for Model Compression and Completion. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2010.

[15] E. B. Sudderth, A. Torralba, W. T. Freeman, and A. S. Willsky. Describing Visual Scenes Using Transformed Objects and Parts. *Int. Journal of Computer Vision*, 77(1–3):291–330, 2008.

[16] Y. W. Teh and M. I. Jordan. Hierarchical Bayesian Nonparametric Models with Applications. In *Bayesian Nonparametrics: Principles and Practice*. Cambridge University Press, 2010.

[17] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet Processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

[18] R. Thibaux and M. I. Jordan. Hierarchical Beta Processes and the Indian Buffet Process. In *Proc. of the Int. Conf. on Artificial Intelligence and Statistics (AISTATS)*, 2007.

[19] R. Triebel, J. Shin, and R. Siegwart. Segmentation and Unsupervised Part-based Discovery of Repetitive Objects. In *Proc. of Robotics: Science and Systems (RSS)*, 2010.