The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Probabilistic Categorization of Kitchen Objects in Table Settings with a Composite Sensor

Zoltan-Csaba Marton, Radu Bogdan Rusu, Dominik Jain, Ulrich Klank, and Michael Beetz
Intelligent Autonomous Systems, Computer Science Department,
Technische Universität München
Boltzmannstr 3, Garching bei München, 85748, Germany
{marton,rusu,jain,klank,beetz}@cs.tum.edu

***Abstract*— In this paper, we investigate the problem of 3D object categorization of objects typically present in kitchen environments, from data acquired using a composite sensor. Our framework combines different sensing modalities and defines descriptive features in various spaces for the purpose of learning good object models. By fusing the 3D information acquired from a composite sensor that includes a color stereo camera, a time-of-flight (TOF) camera, and a thermal camera, we augment 3D depth data with color and temperature information which helps disambiguate the object categorization process. We make use of statistical relational learning methods (Markov Logic Networks and Bayesian Logic Networks) to capture complex interactions between the different feature spaces. To show the effectiveness of our approach, we analyze and validate the proposed system for the problem of recognizing objects in table settings scenarios.**

I. Introduction

This paper investigates the perceptual capabilities of personal robots that are to pick and place objects in the context of everyday manipulation tasks in domestic environments. The objects to be manipulated are those that are typical for kitchen environments and include cups, glasses, plates, silverware, bottles, and boxes. The robots must become capable of detecting, categorizing, recognizing, localizing, and reconstructing the objects that are relevant for their manipulation tasks.

The materials that many kitchen objects are made of present difficult challenges for state-of-the-art sensing technology: glasses and bottles are translucent and knives and forks are shiny. They are therefore difficult to detect. Also, silverware and the bottoms of plates are very flat and therefore difficult to discriminate from the table plane in noisy point cloud data.

Because the perception task is too hard for any individual sensor type, we propose the usage of a composite sensor to recognize and analyze living environment scenes. We show, within a table setting scenario, that we can learn, classify and localize objects like cups, glasses, plates, silverware, bottles and boxes. We are using a set of features that mostly rely on geometric properties, unlike the state of the art methods for such tasks, like the methods discussed in [1], which rely on appearance models.

To perform these perceptual tasks, we use a single composite sensor (see Figure 1) that includes a color stereo camera, a time-of-flight camera, and a thermal camera. A 2D laser sensor is also present in the figure but it is not used for the purpose of the experiments presented in this paper.

Each of the individual sensors cannot obtain enough information for the task of object categorization on its own. Stereo cameras, for example, cannot provide the necessary depth information for untextured planes, glasses are transparent and not easily visible with any camera, and thermal cameras can only provide partial information about the world when the temperature is different. However, combining the complementary perceptual evidence provided by the individual sensors gives us a much more valuable source of information.

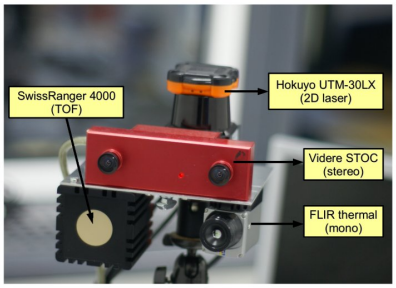

Fig. 1. The proposed composite sensor – please note that the 2D laser is not used in these experiments.

Our composite sensor generates a 3D point cloud $\mathcal{P}$ representation of the sensed scene, where each point $\boldsymbol{p}_i \in \mathcal{P}$ provides the fused information coming from the individual sensors on multiple dimensions, such as the respective color channel values, thermal information, and a distance estimate – to name a few.

The focus of this paper is the sensor data processing pipeline that receives the raw point cloud generated by the composite sensor as its input and returns a categorization of the objects in the sensor view.

The novel aspect of the research reported in this paper is that our composite sensor – together with the proposed sensor data processing pipeline – enables robots to perform object categorization for a range of objects that are typical for domestic environments but represent challenges for state-of-the-art recognition methods. In particular, we are able to perceive and categorize transparent objects such as glasses and bottles, shiny objects such as knives and forks, and untextured boxes.

Therefore, we list our contributions as follows: i) a dual geometric and color processing pipeline for the acquisition of models for shiny and transparent objects in indoor kitchen environments; ii) a fusion of complex feature spaces for the problem of object categorization in the context of perception for manipulation; and iii) the modeling of a machine learning framework able to deal with noisy data (e.g. Time-Of-Flight depth measurements) and partial views.

The remainder of the paper is organized as follows. We address related work on similar initiatives in the next section and describe the system architecture, together with a set of data postprocessing steps in Section III. Section IV briefly addresses the data registration problem using our Visual Odometer (VO), and Section V describes the dual geometric and amplitude segmentation algorithms for extracting the table and the objects supported on it. In Section VI, we present the different feature spaces used for learning the object classes. Section VII outlines our statistical relational learning method, followed by a set of experiments and results for the problem of object categorization in table setting scenarios in Section VIII. Finally, we conclude, giving insights on our future work in Section IX.

II. Related Work

To compute the ego-motion and combine registered views into maps, a combination of a 2D laser, a camera, an inertial measurement unit (IMU) and a stereo pair is used in [2]. The resultant 3D point clouds are obtained using sweeps, which leads to an increased data acquisition time. A method for fusing stereo and time-of-flight (TOF) cameras is presented in [3], where downscaled RGB camera images are calibrated with cropped TOF images. The authors propose a method for enhancing the stereo camera's disparity image, however the depth image from the TOF camera is still better even if it has lower resolution. Another fusion method is shown in [4], where the RGB information is mapped onto the 3D points. In our framework, we annotate the point cloud with RGB information as well, but we additionally use the original RGB image as a source of information for feature extraction, by projecting the 3D mapped clusters into the 2D image and performing the search there.

In [5], [6] two methods are presented for improving the depth image of time-of-flight cameras based on a stereo camera setup and graph-cut and probabilistic algorithms. The cameras used are a 19k type PDM and a SwissRanger 3000. While these methods give promising results, the increased acquisition time for a slight improvement in accuracy is a downside for applications such as ours. A similar approach is presented in [7] but no information is given by the authors on the computational performance of the algorithm.

In our application, the biggest errors in the depth images are produced while scanning shiny or transparent objects, and these are hard to account for using other sensor sources. Therefore, we make use of a machine learning approach for categorization, and use the results to decide how to partially reconstruct the missing 3D data.

Object categorization goes hand in hand with segmentation and is usually performed using a single sensing device. Given a large set of training values containing all possible views, most approaches try to abstract the problem by using features like [8]–[10], which work best on low scale and texture variance. The scaling variance can be reduced significantly by a previous segmentation. [11] tries to extract the 3D world out of only one view in order to improve the segmentation of objects. Another approach is to actively explore the environment and segment objects using the motion to generate 3D shape information [12], [13]. Our approach uses an active TOF camera combined with a small stereo baseline for generating 3D shape features in a still scene for near realtime object segmentation. Additionally, we make use of the 2D images to improve our categorization.

Since we have structured objects with mainly round and straight edges, we propose the use of an ellipsoid feature that is extracted using a method similar to the one proposed in [14]. This approach allows us to match any planar substructure of objects perspectively invariant.

If maps are to be used for more than navigation and mere obstacle avoidance, a semantic interpretation of the observed scenes is required, which necessitates a meaningful labelling of objects appearing in mapped scenes. In [15], the authors propose a two-stage process to solve this problem, where, at the local level, classification is based on appearance descriptors, and at the global scene level, Markov random fields (MRFs) are applied to model relationships. In this paper, we take a similar approach in that we, too, consider a multi-stage process and leverage the promising combination of both TOF data and vision. We, however, advocate the use of statistical relational models [16] such as Bayesian logic networks or Markov logic networks to describe the complex interdependencies in real-world scenes. Such models, by representing general principles about entities and relations, can soundly generalize across arbitrarily complex situations with varying numbers of interrelated objects, and they subsume graphical models such as MRFs. In the literature, approaches that consider segmentation and classification at the same time – by defining an MRF directly over the scan points – have also been proposed [17], yet we believe that a hierarchical approach is preferable, not only from the perspective of computational efficiency. In [18], the authors draw upon statistical relational learning methods, specifically associative Markov networks, to solve a similar problem – again, however, on laser data only and at a low level of abstraction.

III. System Architecture and Data Postprocessing

The overall architecture of our system is depicted in Figure 2. The individual processing steps are depicted as rectangles with sharp corners while the data structures are shown as rectangles with round edges. The input shown at the top of the figure are the image streams provided by the components of the composite sensor: the left-right streams of the color stereo camera, the infrared image stream, and

the distance camera images produced by the time-of-flight camera. The output, depicted at the bottom comprises a set of reconstructions of the objects in the scene together with their categorization.

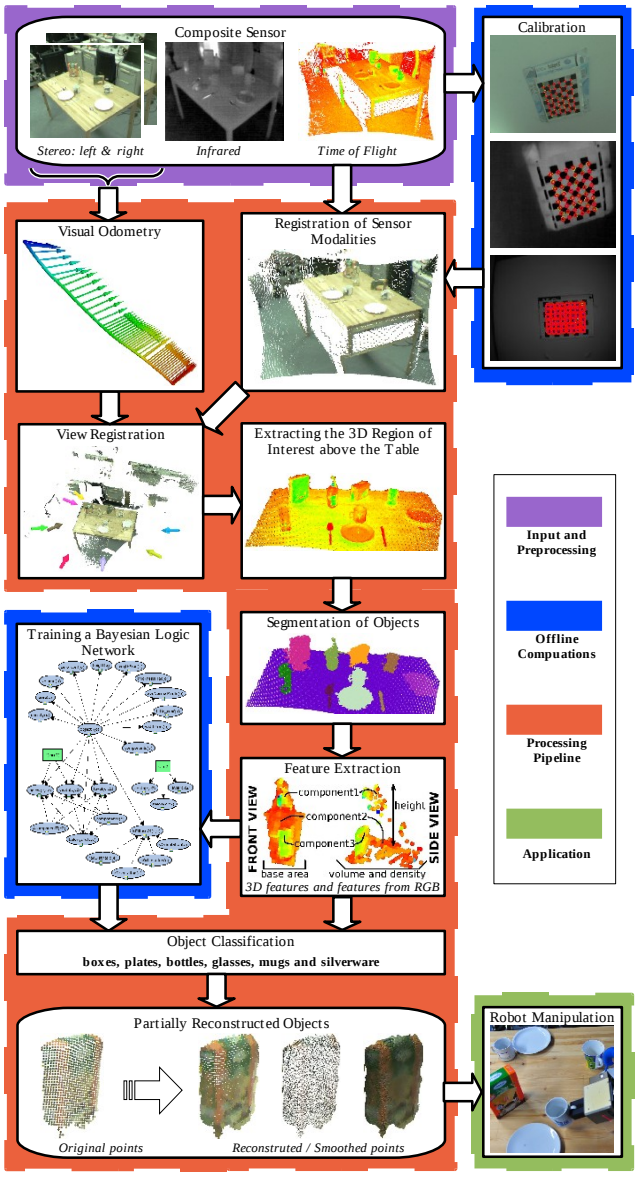

Fig. 2. An architectural overview of our processing system.

A preliminary offline step for an accurate data fusion is the calibration of the different sensing devices with respect to each other. The calibration enables the data interpretation algorithms to solve the sensor data correspondence problem and to bring the images in the same coordinate system. Since the cameras are rigidly mounted, we assume that the extrinsic calibration parameters do not change. In our setup, we calibrated every two cameras as a stereo pair. The resultant pixel backprojection errors were kept below 10%. As a future extension we plan to build a better calibration plate and investigate the overall sensor calibration procedure by treating the cameras as a tri- or quadrifocal sensor.

The first processing step in our data processing pipeline is the *registration of sensor modalities* step. In this step the sensor data of the individual sensors are combined to form a 3D point cloud representation of the sensor view. Each point in the cloud carries information from all sensors. Stereo depth estimates are typically very accurate at edges but completely missing for untextured planes. The depth estimates provided by the time-of-flight camera are typically very noisy at occluding edges in the scene.

The *view registration* step registers the individual point cloud views into a global point cloud. This step is supported by the *visual odometry*, which estimates the poses of the camera relative to the scene based on the stereo image stream obtained so far. The pose estimate of the camera, together with the relative viewpoint of the cloud, provides the solution for the new view's integration into the global 3D point cloud.

Next, an extraction of the 3D regions of interest is performed. In the example depicted in Figure 2, the robot is to interpret a table-setting scene. Therefore, the table needs to be segmented and the objects on top of it must be clustered and extracted as separate regions.

The *segmentation of objects* step will eliminate points that are generated by the table itself in order to find the sets of points that correspond to objects lying on the table. These sets are then segmented to form hypotheses about objects on the table. For each of them, the object feature inference mechanism infers the respective feature values.

The *object categorization* is then performed by phrasing it as a probabilistic inference task based on learnt statistical relational models about objects, their composition and their perceptual features. To this end, we have trained a Bayesian logic network that represents the relationships between features and object categories as a graph and then learns the probabilistic correlation between objects, categories, and their perceptual features from a set of training examples. The categorization step returns, for each perceived object, the most probable category and a measure that signals how confident the system is that the categorization is right.

Finally, using the information provided by the categorization system, the *object reconstruction step* partially reconstructs object surfaces into more compact models by the use of triangular meshes. Currently we only reconstruct objects that provide a reliable geometric support in the point cloud data, like boxes, as presented in [19]. However, we plan to extend our system to create different reconstruction methods for different object classes (e.g. use a cylinder to represent glasses and bottles).

The following sections describe the aforementioned processing steps in more detail.

IV. Partial-View Registration

Instead of aligning partial point cloud views in a rigid registration framework, our solution employs a Visual Odometer on the stereo camera images. More precisely, for each series of monocular and disparity images from the stereo camera, the VO computes the camera motion between the views and estimates a rigid transformation (3D rotation and translation).

While recent work [20] has addressed the problem of estimating the camera motion using a SwissRanger SR-3k TOF camera, this did not give satisfactory results for our setup where the amplitude of the image is varying substantially depending on whether the camera is closer or further away from the object scene. Therefore, our Visual Odometry system [21] seeks to estimate the camera motion by tracking CenSurE (Center Surround Extrema) features from one stereo image pair to the other. Then, the motion is scored using the pixel reprojection errors in both cameras, and its inliers are evaluated using the disparity space homography. The hypothesis with the best score (maximum number of inliers) is used as the starting point for a nonlinear optimization routine. Figure 3 presents the fusion of two different point cloud views using the camera motion estimates given by the VO framework.

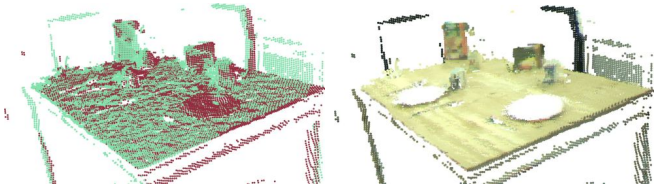

Fig. 3. Left: the fusion of two different point clouds acquired from separate views using the information provided by the Visual Odometry system. Right: the mapping of RGB information to the resultant global point cloud model.

V. Support Planes and Object Segmentation

For a partial view or a given set of partial views registered together, our framework proceeds to extract horizontal planes which might support objects on them.

The supporting planes are computed by making use of standard robust estimators such as RANSAC [22] and fitting restricted planar models to it. In our implementation, we assume that the camera is close enough to the actual table and looking at it, so the largest planar model found closest the viewpoint is assumed to be the table. Once a model has been found, we compute the table inliers $\mathcal{T}$ and boundaries, approximate them with a convex polygon, and extract all point clusters located on it by checking if their projection on the plane intersects the polygon. This results in a rough geometrical segmentation of the scene.

An important aspect of the sensed 3D data is that because of highly reflective or non-refractive surfaces, points which physically describe a given object will be estimated and sampled somewhere else in space, thus returning erroneous measurements. This mostly occurs with glasses, bottles, and silverware, but can seldom be encountered for certain cups as well. Though in most cases these erroneous points are clustered together in space above the table plane, there are situations where they are distributed along the table surface, thus making it impossible for any geometric method to segment them accurately (see Figure 4).

A solution to this problem is to perform a clustering of the point cloud in the SwissRanger amplitude space, which can be used together with the geometric information to extract different object components. Our clustering method is based on a region growing approach, where points with the same amplitude characteristics are added to a list of seed points, and a region is grown until no neighboring points with the same characteristics are left. Since it is impossible to set the amplitude clustering thresholds to some fixed values that would work for any table setting, we automatically estimate them as follows.

We approximate the distribution of the amplitude values of the points in $\mathcal{T}$ with a Gaussian distribution ($\approx 65\%$ being in the $\mu \pm \sigma$ interval) and we compute the mean μ and standard deviation σ. Then, we estimate the mean $\overline{\mu}$ of the amplitude differences between neighboring points in $\mathcal{T}$ as

$$\overline{\mu} = \frac{1}{nr_T} \sum_{t_i \in T} \left(\frac{1}{nr_N} \sum_{n_j \in \mathcal{N}_i} \left| t_i^{int} - n_j^{int} \right| \right) \qquad (1)$$

where $\mathcal{N}_i \subset \mathcal{T}$ is the set of neighboring pixels of t_i in the amplitude image that are also on the table ($n_j \in \mathcal{T}$), t_i^{int} and n_j^{int} are the amplitude values at the respective points, and the number of points in $\mathcal{T}$ and $\mathcal{N}_i$ is nr_T and nr_N respectively.

Since the goal is to prevent the undersegmentation of the table and the objects on it, we can tighten the statistical threshold and use $\overline{\mu}/2$ as a connectivity criterion in amplitude to limit the 3D region growing in the table points. Additionally we impose a maximum divergence of two standard deviations to avoid growing too far even if there is a small gradient. This way we obtain multiple region patches on and in the vicinity of the table.

Fig. 4. A table setting view in amplitude space (grayscale). A close-up of a glass object is shown in the left part of the figure, showing no geometry information at all. For a side view including the bottle see Figure 3.

Due to the strict thresholds, the different parts of the table will be oversegmented, but they can be easily merged by verifying their estimated plane normal to the one estimated for all the points in $\mathcal{T}$. The rest of the points are marked as either belonging to an object on the table (e.g. a plate) or to a *shadow*. The parts of the scene where the amplitude values change drastically will produce very small regions. These *shadow* points are typically produced either by shiny objects like silverware or transparent objects that create patterns on the part of the table which they occlude. Glasses and bottles in particular reflect the waves of the sensor such that some of the resulting points will be below or inside the table (see Figure 4, for example).

By using the relative viewpoint information of every scene, we can locate the points that are above or slightly below the

table and group them into connected components – labeled as belonging to objects outside the table's plane.

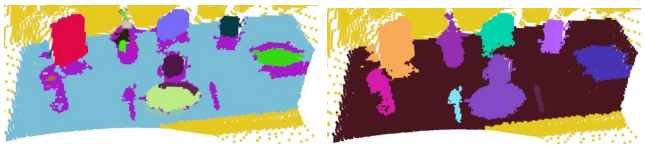

Fig. 5. Left: different object regions in different colors together with points on the table (light blue) and discarded points (yellow). Note that shadow points are in very small regions and are marked with the same label (purple). Right: object candidate clusters in random colors. (For interpretation of the references to color in this figure legend, the reader is referred to the online version of this article.)

Thus, our region growing method will yield three types of point labels: *outside* – belonging to a region of an object that is outside the table, *near* – belonging to a region of an object that is near the table, and *shadow* – belonging to a small region with distinctive amplitude value near the table. The resulting labeling can be seen in the left part of Figure 5, where each connected group of points with the same label is considered a component of an object. These labeled points are then clustered together in 3D to form object candidates (see Figure 5, right) and a list of features are extracted for each of them, as detailed in the next section.

VI. Feature Spaces

Each cluster representing an object candidate is analyzed based on the different sensor modalities and the resulting features are fed to the *object categorization*.

A. 3D Geometric Features

By analyzing the points in each cluster we can extract the features described below, based on the measurements of the time-of-flight camera. The features can be grouped into two categories: *attributes* – atomic values, and *relationships* – features computed by comparing parameters of the clusters' components (see *feature extraction* in Figure 2).

The list of computed attributes is as follows:

- *average normal angle*: the average angle of the estimated point normals (relative to the table's normal);
- *maximum and average height*: maximum and average distance of the points from the table's plane;
- *base area*: estimated area of the cluster's footprint on the table, based on occupancy of octree leaves;
- *volume*: estimated volume of cluster, based on the maximum heights above the cells in the base area;
- *average point density*: number of points in unit volume[1];
- *thickness, longness, wideness*: proportions of cluster along its principal directions;
- *points above and below the table*: percentage of points with positive / negative distance to the table's plane;

[1]An object which has shiny or transparent parts (like a bottle) will have a considerably lower density than the normal variations one gets by changing distances in indoor applications (especially considering the sensor limits too). The clustering in Section VII differentiates between the two cases.

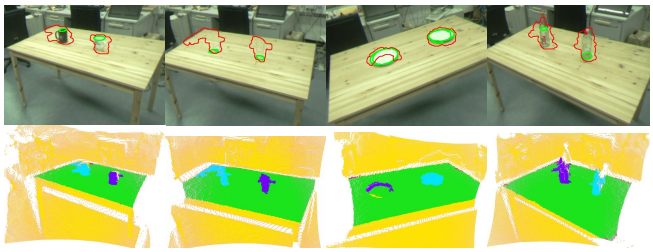

Fig. 6. Training samples for the classes mug, glass, plate and bottle. The upper row shows the RGB images with the 3D Segmentation projected and dilated with corresponding detected ellipses in green.

- *shadow and near points*: percentage of points marked with the *shadow* and *near* label;
- *zero and low confidence*: percentage of points which have 0 and respectively very low ($< 1\%$ of the maximum) confidence values assigned by the sensor [2];
- *number of components*: the number of *outside*, *near* and *shadow* connected components defined as detailed in the previous section.

The relationships which we defined between the components (if there are more than one components in the cluster):

- *above / below / same level*: the order of the components based on their centroid's height relative to the table;
- *front/behind* the order of the components based on their distance from the viewpoint.

B. 2D Color-Camera analysis

We start analyzing the RGB images of the stereo camera with the 3D segmentation of the data giving a set of corresponding pixels for each cluster. Since the resolution in 3D is lower than that of the camera, these pixels have to be dilated in order to obtain the clusters in the camera images. These image segments are then searched for occurrences of ellipses, which are then characterized by a quantization describing their orientation relative to the table and their size.

By searching for elliptical elements that appear on round or cylindrical structures, we want to improve the detection of mugs, plates, glasses and bottles. To reach the goal of matching elliptical structures, the approach of Hofhauser et al. [14] is applied on a circular edge-model. This method matches projective distorted edge templates in an image. It extracts the edges on several pyramidical levels of an image and searches for possible occurrences of the template on the highest pyramid level (lowest resolution) and tracks candidates through the pyramid. All edge pixels of a model image are used to create such a template, which is then divided into several sub-structures using simple clustering. It is assumed that under perspective distortion, small structures do not change significantly. The final matching in the images is based on the a variation of the metric proposed in [23], with some extensions to also support projective transformation of the models.

[2]The SR-4k grades each measurement accuracy with a value, 0 being the lowest confidence, typically assigned to points on metallic surfaces.

The approach allows to detect most of the visible circles and returns the relative position depending on the actual radius of the ellipse. Some objects have circular parts that are partially visible in an edge image, and most of these partial ellipses will also be found by the approach we apply. To filter out false positive matches in the clutter, we allow only circles that would lay on the table. Those circles will be used as features with their radius-distance ratio and their rotation against the camera. 3D segments and objects are connected by restricting the search to the projected 3D point cloud that is segmented as an object. The radius-distance ratio can be computed by assuming that the ellipse is the projected image of a circle. Such an ellipse defines two possible poses for that circle. These poses differ only in orientation and have, for any circle radius, a fixed distance. This distance is indirectly proportional to the radius and defines a unique ratio. Figure 6 (upper row) shows ellipses extracted in the training data. The green ellipses are assigned to the containing segment as an additional feature.

C. Extracting Temperature Information

At this stage, we are interested in the temperature of an object (or its content), i.e. whether it is hot, room temperature or cold. We compute the average of the temperature readings, and compare it to room temperature (taken from the temperature of the table). This makes sense especially for mugs, bottles, glasses and possibly plates, as it can provide information about content or the lack of it. We intend to experiment with the possibility of using this data for improved action planning.

VII. Object Categorization

We use state-of-the-art statistical relational learning methods in order to classify the entities that were previously segmented. In statistical relational models, we can capture complex interactions between objects, their attributes and relations between objects. By describing general principles about multiple objects having similar properties, i.e. about classes of objects, the models can be applied to arbitrary situations with varying numbers of objects, simply by utilizing the subset of principles that applies to them. A statistical relational model can thus be thought of as a template for the construction of a probabilistic graphical model, in which one can perform inference using standard methods.

Our framework supports two rather complementary representations, Markov logic networks (MLNs) [24] and Bayesian logic networks (BLNs) [25], which can be regarded as a dialect of multi-entity Bayesian networks [26]. While MLNs are generally more expressive, learning is, unfortunately, an ill-posed problem [**?**], and both learning and inference tend to be more computationally expensive than in BLNs. Since we here do not require the added expressiveness, and MLNs otherwise offer few benefits apart from the support for discriminative learning, we opted for BLNs.

A BLN $\mathcal{B} = (\mathcal{D}, \mathcal{F}, \mathcal{L})$ is a relational model in which the variables under consideration are first-order terms or predicates. A model consists of a set of type, predicate and function declarations $\mathcal{D}$ indicating the types of the entities to which the predicates/functions are applicable, a set of fragments $\mathcal{F}$ specifying conditional probability distributions (CPDs) which are applicable to a random variable under certain circumstances, and a set of hard logical constraints $\mathcal{L}$ formulated as sentences in first-order logic. For any given set of (typed) entities E, we obtain, in accordance with the set $\mathcal{D}$, a set of random variables which constitutes the set of nodes in a ground (auxiliary) Bayesian network $B_{\mathcal{B},E}$ specific to E, which is obtained by materializing the template structures defined in $\mathcal{F}$ appropriately.

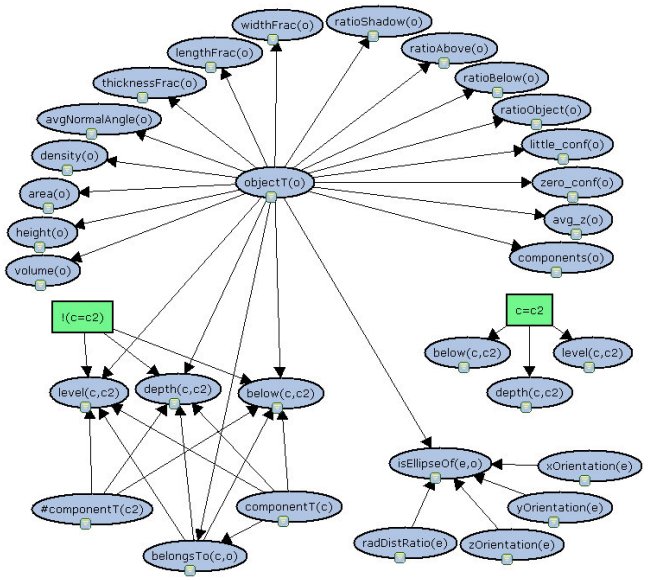

Fig. 7. The model structure with dependencies as conditional probability distribution fragments.

A statistical relational model is, in general, quite well-suited to the categorization task that we address in this paper. An object may consist of a variable number of spatially related components, and the relational model can potentially consider an arbitrarily complex configuration of components, sensibly adjusting its prediction with each new component and relationship it is told about. Moreover, an object may or may not be associated with optical features such as ellipses, and our relational models can deal with the presence of an arbitrary number of such features. For our classification task, we thus modelled the probabilistic dependencies between an object's class, its immediate attributes, the typed components that belong to it and the spatial relationships between them, as well as ellipses that were detected on the objects and parameters of these ellipses. The corresponding model structure is shown in Figure 7 (where the oval nodes indicate generalized random variables and the rectangular nodes indicate preconditions for the respective fragments to be applicable). Since our domain lacks global logical constraints, the set $\mathcal{L}$ is empty in our model.

Because BLNs in their current form are strictly discrete, we discretized the continuous 2D and 3D features mentioned above using expectation maximization clustering as implemented in WEKA [27], which resulted in domain sizes rang-

ing from two to ten, depending the variability in the training set. Generative training of BLNs is particularly simple, as it essentially reduces to the computation of relative frequencies in the data. For inference, our probabilistic framework supports state-of-the-art Bayesian network inference methods, yet for our concrete problem, even exact inference is still feasible. For any concrete set of entities E, the ground Bayesian network B_E represents a full-joint distribution over the random variables implied by the template structure and E, which we can use to infer arbitrary conditional probabilities. In our case, we are interested in conditional probability queries of the form

$$P(\text{objectT}(O) \mid e, B_E) \tag{2}$$

where O is one of the segmented objects we wish to classify and e is the conjunction of observations that were made about O, i.e. a conjunction such as

$$\textit{height}(O)\text{=}\textit{Height2} \wedge \textit{hasComponent}(O, C_1) \wedge \textit{componentT}(C_1)\text{=}\textit{Shadow} \wedge \textit{isEllipseOf}(E_1, O) \wedge \ldots$$

In domains where the objects to be classified have meaningful arrangements (i.e. their positions relative to each other are not arbitrary), we could even consider spatial relations between them and consequently perform *collective categorization*, i.e. we could collectively determine the most likely categorization of an entire group of objects, taking interactions between neighboring objects into consideration. Collective classification was, for instance, done based on the relational Markov network framework in [18] to identify walls, windows and gutters in laser data. In our domain, the inclusion of inter-object relations would in principal be feasible, however the nature of our training data was such that no meaningful dependencies could have been extracted.

VIII. Experimental Scenario and Discussions

To evaluate our approach, we applied our system to the problem of object categorization in table-setting scenes. The types of objects include regular everyday kitchen objects, such as boxes of cereals, boxes of tea, different types of glasses and mugs, a few bottles, plates, and two types of silverware. The training data for the categorization was generated by placing two objects of the same type on the table, and creating a series of views in a 180° arc of a circle, which were then processed through our geometric segmentation and feature estimation pipeline. A few examples of such scenes are presented in Figure 6, with the RGB images and the ellipsoidal features in the first row, and the 3D point clouds and the segmented clusters in the bottom row.

We generated roughly 50 views for each object class which resulted in ≈ 80 training examples for bottles, boxes, glasses, mugs, plates and silverware. From some viewpoints, occlusions or measurement errors produced incorrect object clusters, leading to errors in the class label estimation. These cases were marked as *ambiguous*, and we trained the model by including a few of these cases, since they do occur in highly cluttered scenes like the one we used for testing. Please note that while we didn't train for all the special cases

Class	Correct	Number	Ratio
1 - box	23	27	0.85
2 - plate	52	67	0.78
3 - glass	7	31	0.23
4 - mug	39	40	0.98
5 - bottle	4	5	0.80
6 - silver	21	43	0.57
7 - ambiguous	17	87	0.20
overall (1-7)	163	300	0.54
objects (1-6)	146	213	0.69
comparisons			
no relations between components	143	300	0.48
no 2D data	151	300	0.50
no 3D data	62	300	0.21
decision tree	148	300	0.49

TABLE I

Categorization results for all experiments performed (see Section VIII for details).

ground truth \ categorization	box	plate	glass	mug	bottle	silver	ambiguous
box	23	0	0	2	1	0	1
plate	0	52	0	0	0	5	10
glass	0	14	7	0	0	2	8
mug	0	0	0	39	1	0	0
bottle	1	0	0	0	4	0	0
silver	0	0	7	2	0	21	13
ambiguous	20	8	1	5	28	8	17

TABLE II

Confusion matrix for the results in Table I.

on how objects can occlude each other, the learned model still recognized some unseen examples of clustering failure. However, we plan to address the respective shortcomings in the near future.

Our training database contained 19224 facts on 570 objects that were comprised of 1851 spatially related components and featured 1308 ellipses. Training was completed in roughly 10 minutes. The learned model was applied to a more complex, partially cluttered scene (see Figures 3 and 8), which contained a mixture of new unseen objects with random objects picked from the training dataset.

Table I shows the classification results, which indicate an overall classification rate of about 54%, yet the accuracy on properly segmented objects is almost 70%. The time taken for a run of the feature estimation and the classifier on a single scene was but a few seconds.

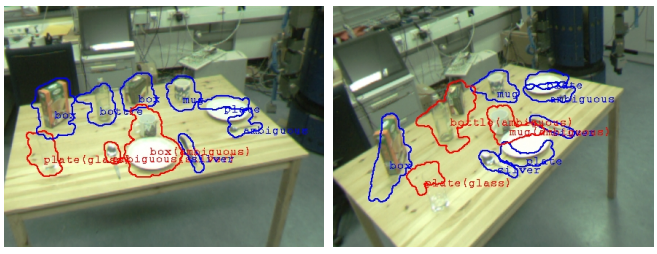

Fig. 8. Categorization results for the partial view dataset presented in Figure 4 and of the same scene from a different view. Correctly classified clusters are marked with blue, while incorrectly classified ones with red, and the ground truth is shown in parenthesis.

During the experiments we observed that the ellipsoidal features helped with the categorization of plates, and most of the mugs and glasses. Due to low contrast in the scene, the system extracted a few ellipsoid features on some of the spoons or several textured parts of the scene, which usually lead to false positives. However, because of the probabilistic nature of our learning scheme and the number of feature spaces we used, the model was not affected significantly. We plan to investigate this further in the future, and include additional visual features. Additional features would also help distinguish the glasses better from other types of objects (plates in particular) and improve their classification rate.

The ambiguous category was also recognized only 20% of the time, but in some cases, our system recognized such ambiguous clusters instead as one of the object classes that do indeed appear within the cluster, even though the presence of clutter was not explicitly trained for. Such a case can be seen in the right part of Figure 8 (the mug touching part of the plate and the bottle partially occluding the box).

IX. Conclusions and Future Work

We have presented a system that can extract features from different sensor modalities for solving the problem of classifying different objects present in kitchen environments. Our experimental results indicate that the fusion of sensory information is indeed helpful, since any model that considered only a subset of the information that was available failed to deliver equivalent results (see bottom part of Table I).

The accurate categorization of objects in mapped scenes is a crucial step in the reconstruction of partially missing 3D data. For objects which are known to be made out of materials that produce relatively accurate TOF camera readings, our reconstruction method is already capable of smoothing out sensor noise while preserving sharp features – like the edges of boxes – as much as possible and of producing a triangular mesh representation (see Figure 2 bottom). For shiny and transparent objects, however, we plan to investigate the use of other sensor modalities and the modeling of the TOF measurements of these surfaces.

Acknowledgements

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

References

[1] P. Moreno, M. Marín-Jiménez, A. Bernardino, J. Santos-Victor, and N. Blanca, "A comparative study of local descriptors for object category recognition: SIFT vs HMAX," *Lecture Notes in Computer Science*, vol. 4477, p. 515, 2007.

[2] G. B. Mirisola, J. Lobo, and J. Dias, "3D map registration using vision/laser and inertial sensing," in *European Conference on Mobile Robots (ECMR2007), Freiburg, Germany, Sep.*, 2007.

[3] S. A. Guomundsson, H. Aanæs, and R. Larsen, "Fusion of Stereo Vision and Time-of-Flight Imaging for Improved 3D Estimation," in *International workshop in Conjuction with DAGM'07: Dynamic 3D Imaging*, Sep 2007.

[4] L. M., K. A., and H. K., "Data-Fusion of PMD-Based Distance-Information and High-Resolution RGB-Images," in *International Symposium on Signals, Circuits and Systems (ISSCS)*, July 2007.

[5] H. Uwe and A. Marc, "Combining Time-Of-Flight depth and stereo images without accurate extrinsic calibration," *Int. J. Intell. Syst. Technol. Appl.*, vol. 5, no. 3/4, pp. 325–333, 2008.

[6] J. Zhu, L. Wang, R. Yang, and J. Davis, "Fusion of Time-of-Flight Depth and Stereo for High Accuracy Depth Maps," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[7] J. Diebel and S. Thrun, "An application of markov random fields to range sensing," in *Proceedings of Conference on Neural Information Processing Systems (NIPS)*. Cambridge, MA: MIT Press, 2005.

[8] T. Quack, V. Ferrari, B. Leibe, and L. Van Gool, "Efficient Mining of Frequent and Distinctive Feature Configurations," in *IEEE 11th International Conference on Computer Vision (ICCV)*, 2007, pp. 1–8.

[9] P. Yan, S. Khan, and M. Shah, "3D Model based Object Class Detection in An Arbitrary View," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, 2007, pp. 1–6.

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

[11] A. Saxena, M. Sun, and A. Ng, "Learning 3-D Scene Structure from a Single Still Image," *IEEE 11th International Conference on Computer Vision (ICCV), 2007.*, pp. 1–8, 2007.

[12] K. Welke, T. Asfour, and R. Dillmann, "Object separation using active methods and multi-view representations," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2008, pp. 949–955.

[13] D. Feldman and D. Weinshall, "Motion Segmentation and Depth Ordering Using an Occlusion Detector," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 7, pp. 1171–1185, 2008.

[14] A. Hofhauser, C. Steger, and N. Navab, "Harmonic deformation model for edge based template matching," in *Third International Conference on Computer Vision Theory and Applications*, vol. 2, 2008, pp. 75–82.

[15] I. Posner, M. Cummins, and P. Newman, "Fast Probabilistic Labeling of City Maps," in *Proceedings of Robotics: Science and Systems*, Zurich, June 2008.

[16] L. Getoor and B. Taskar, *Introduction to Statistical Relational Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2007.

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

[18] R. Triebel, K. Kersting, and W. Burgard, "Robust 3D scan point classification using associative markov networks," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2006, pp. 2603–2608.

[19] Z. C. Marton, R. B. Rusu, and M. Beetz, "On Fast Surface Reconstruction Methods for Large and Noisy Datasets," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Kobe, Japan*, 2009.

[20] S. May, D. Droeschel, D. Holz, C. Wiesen, and S. Fuchs, "3D Pose Estimation and Mapping with Time-of-Flight Cameras," in *International Conference on Intelligent Robots and Systems (IROS), 3D Mapping workshop, Nice, France, September 22-26*, 2008.

[21] B. Morisset, R. B. Rusu, A. Sundaresan, K. Hauser, M. Agrawal, J.-C. Latombe, and M. Beetz, "Leaving Flatland: Toward Real-Time 3D Navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Kobe, Japan, May 12-17*, 2009.

[22] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," in *Comm. of the ACM, Vol 24, 1981*, 1981.

[23] C. Steger, "Occlusion, clutter, and illumination invariant object recognition," in *International Archives of Photogrammetry and Remote Sensing*, vol. XXXIV, part 3A, 2002, pp. 345–350.

[24] M. Richardson and P. Domingos, "Markov Logic Networks," *Mach. Learn.*, vol. 62, no. 1-2, pp. 107–136, 2006.

[25] D. Jain, S. Waldherr, and M. Beetz, "Bayesian logic networks," IAS group, Technische Universität München, Fakultät für Informatik, Tech. Rep., 2009.

[26] K. B. Laskey, "MEBN: A language for first-order Bayesian knowledge bases." *Artif. Intell.*, vol. 172, no. 2-3, pp. 140–178, 2008. [Online]. Available: http://dblp.uni-trier.de/db/journals/ai/ai172.html#Laskey08

[27] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations (The Morgan Kaufmann Series in Data Management Systems)*, 1st ed. Morgan Kaufmann, October 1999. [Online]. Available: http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/1558605525