

Perception and Probabilistic Anchoring for Dynamic World State Logging

Nico Blodow, Dominik Jain, Zoltan-Csaba Marton, Michael Beetz
{blodow,jain,marton,beetz}@cs.tum.edu

Abstract—Knowing precisely where objects are located enables a robot to perform its tasks both more efficiently and more reliably. To acquire the respective knowledge and to effectively use it as a resource, a robot has to go through the world with “open eyes”. Specifically, it has to become environment-aware by keeping track of where objects of interest are located and explicitly represent their geometrical properties. In this paper, we propose to equip robots with a perception system that passively monitors the environment using a 3D data acquisition system, identifying objects that might become the subject of future manipulation tasks. Our system encompasses a 3D semantic mapping and reconstruction pipeline and a storage and data merging unit for perceived information that provides on-demand modeling and comparison capabilities. Based on probabilistic logical models, we address the important perceptual subtask of object identity resolution, i.e. inferring which observations refer to which entities in the real world (perceptual anchoring). Our system can be used as a bootstrapping system for the generation of object-centric knowledge and can, in this way, be used as a mid-level perception system that enables activity recognition, scene recognition and high-level planning.

I. MOTIVATION

Robots operating in human environments have strong demands on their perception systems, as they must be able to deal with manipulable objects, containers, doors and tables, and they must, in particular, be able to deal with dynamic environments and with a high degree of variability induced by humans, i.e. objects (dis-)appearing, doors being opened and closed, etc. Consider a scenario where the human asks the robot to retrieve a certain object. Many systems use a task-directed approach to this problem, such that the user query would result in a (possibly time-consuming) search for the object using the sensory inputs available to the agent. However, if the robot had previously observed the object in question, it should simply be able to *remember* where it was (and is still likely to be).

Ideally, the robot’s memory should be able to locate objects that are in plain sight or were observed recently and be able to merge information about object geometry, appearance and common locations of an object as well as a history of where it was previously located into a long-term memory that continuously logs the state of the world. This is beneficial for a large number of open robotics research topics, like grasping (which requires 3D information about the object as well as its surroundings for motion and grasp planning as well as collision avoidance) or higher-level reasoning (e.g. inferring human actions and activities from observations).

Another important aspect is the fact that human environments do not always contain uniquely identifiable objects,

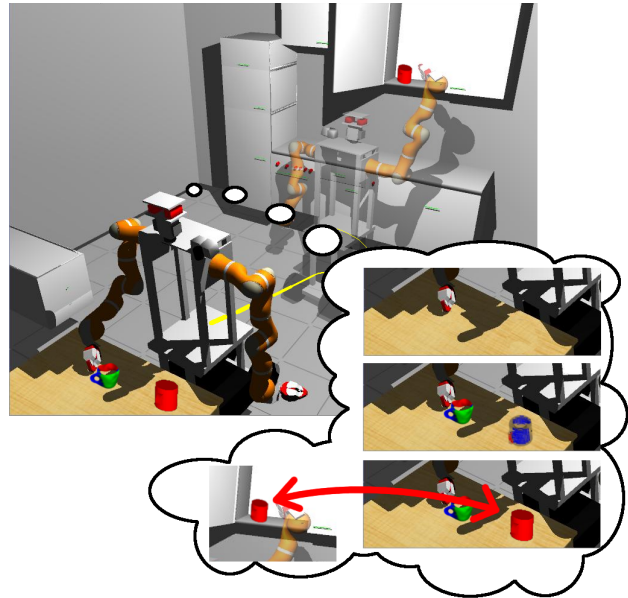


Fig. 1: Keeping track of objects over time: A robotic household assistant must know about its environment and the objects within – both in the temporal and spatial domain.

but rather sets of identical cups, plates and the like. However, such objects should not necessarily be treated as interchangeable. For example, over the course of a meal, a certain cup might be associated with exactly one person, and the robotic assistant should not equate it with all other cups.

This problem gets more difficult if one considers the case where objects can be hidden (e.g. behind cupboard doors) or outside the robot’s view – especially if not all objects are known beforehand. We believe that the robot has to maintain a belief state about (temporarily) unobserved objects and assume that they are likely to remain at their locations unless there is evidence to the contrary. If an object is uniquely identifiable (e.g. a teapot), this problem is trivial (one can simply overwrite its old position), but if there are two or more objects with equivalent appearance, we must maintain more than one hypothesis. If we saw a certain cup earlier in an area that is currently unobserved and we now perceive a similar cup elsewhere, we must consider the two possibilities that a) the two observations are associated with the same object (which therefore moved), or that b) the former cup remained at its position and we are in fact now perceiving a different cup.

We therefore propose a *dynamic object store*, i.e. a system that processes the streams of geometrical information coming in via the robot’s sensors, reconstructs and identifies the

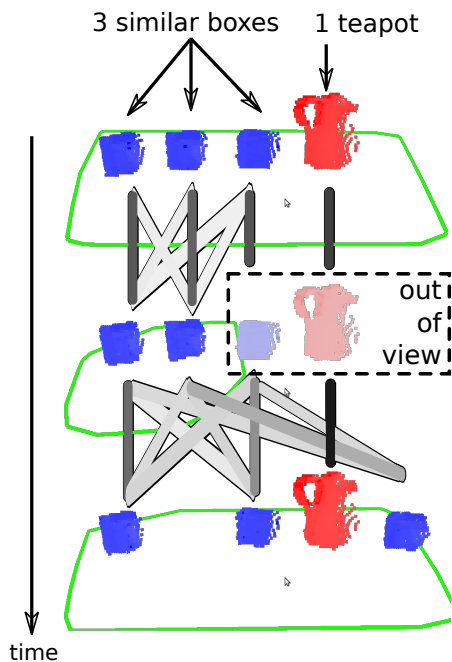


Fig. 2: Illustration of the proposed system’s output (boxes and teapot clusters detected on a table) over three time steps.

observed objects and aggregates the relevant data about them. The system maintains a probability distribution representing its beliefs about object positions and identities (including objects in unobserved areas). The system furthermore mediates between the data acquisition and high-level planning layers. It takes care of the processing of sensor data, the construction of geometric and appearance-based models as well as the probabilistic tracking and logging of objects. Since observations are at first obtained as raw data, the model refinement and classification methods need to be integrated into the system as well.

Our main contributions with respect to the advancement of autonomous behaviour include

- a dynamic object store, which realizes an extendable way of integrating the given sensing modalities with processing routines and reasoning modules,
- tackling the problem of identity management in a probabilistic manner while maintaining a belief state of areas that have been observed previously and are (temporarily) unobserved, and
- enabling spatio-temporal reasoning by logging the state of the environment in a passive manner over longer periods of time.

Figure 2 illustrates both the problem statement and the resulting output of the algorithms described in this work. It depicts three consecutive time steps of a table scene where 3 teaboxes and a teapot are being placed and moved on a table. Note that the colored parts were directly taken from screenshots of the passive perception pipeline visualizer and originate from real laser scan measurements. They are overlaid with the results of the object identity resolution step as a set of connecting lines. The lines’ grey levels represent the probabilities of the two involved observations

originating from the same real-world object, as explained in Section V. The scenario in Figure 2 is very similar to those discussed further on, the only difference being that the objects were placed on a single table to facilitate visualisation.

In the next section, we give an overview of related work. We proceed with an overview of our system (Section III), which includes descriptions of the passive perception pipeline and of the integrated refinement methods. Next, in Section IV, we present our probabilistic framework for object identity resolution, which is based on Markov logic. We provide experimental results in Section V and conclude with Section VI.

II. RELATED WORK

To the best of our knowledge, a system as proposed in this work has not been previously suggested in the literature, yet some of the subproblems we address have been previously tackled.

In [1], the authors propose a memory assistance tool that tracks objects in an environment that is surveyed by cameras to help people with memory loss. Unfortunately, the system is not able to treat non-uniquely identifiable objects, and there is no notion of unobserved areas.

Learning spatio-temporal structures has been employed by research in visual analytics for the analysis of movement data. In [2], behavior patterns are learned from GPS-data. Data mining tasks are performed by human experts using visual analytics methods. Others [3] perform some of these learning tasks using probabilistic learning methods. The acquisition of such models is also investigated in the pervasive systems community ([4], [5]), where models of daily activities are learned from ubiquitous sensor networks. We are proposing a system that passively and unobtrusively acquires the underlying data needed for these approaches, and provides mid-level information about the positions of objects over time.

A key problem that has not received substantial attention in autonomous robotics yet is the inference about whether two observations refer to the same real-world object. This type of inference is needed to recognize previously observed objects as being the same entity, which in turn must be solved to build spatio-temporal environment models. The entity resolution problem arises in several information integration scenarios. Early work has often phrased the problem as a classification problem, where a pair of records would be independently classified as either matching or non-matching [6], more recent approaches have phrased the problem as link prediction in statistical relational models [7]. However, we are not aware of any robotics or computer vision system where such a problem has been explicitly treated. Given the temporal nature of the object identity resolution task, the problem is related to multi-hypothesis object tracking [8], [9] and statistical data association [10]. However, our belief update process operates at fairly low frequencies (rendering the types of motion models usually used in object tracking inapplicable) while incorporating a multitude of logical as well as probabilistic constraints that call for a highly expressive probabilistic logical formalism.

Filtering in purely logical domains was previously considered in [11].

In most computer vision-based approaches, systems rely on features like the ones described by Lowe [12] or Lepetit and Fua [13] to find matches between regions in an image and a database of object images. These approaches, however, stop at this point, assuming that the respective features suffice to uniquely identify an object. We believe that this is an unrealistic assumption and therefore propose a probabilistic framework that allows us to integrate several distance metrics based on descriptors and geometric models (based on 3D mapping), and to reason about object identity and similarity.

The methods proposed in this paper rely on a number of geometric processing algorithms that were presented in previous publications by the authors. Acquiring a semantic object map using 3D sensing is done by first segmenting, then classifying the relevant regions as in [14]. This approach has the advantage of providing labeled and complete reconstructed models for the furniture pieces that are of interest to a household assistant.

III. DYNAMIC MAPPING SYSTEM

Our dynamic mapping system consists, at its core, of the *passive perception pipeline*, which continuously collects and computes information about objects from 3D laser data. This information is subsequently stored in the *dynamic object store* (see Figure 3) and is refined in order to find better representations of the objects represented in the data. We use specialized algorithms to detect boxes and rotationally symmetrical objects [15]. In the case of irregular shapes, we fall back to a hybrid representation (encompassing shape primitives combined with triangulated meshes) [16]. In the following, we describe the passive perception pipeline and our set of refinement algorithms in more detail.

A. Passive Perception Pipeline

The passive perception pipeline consists of:

- *point cloud processing*, which assembles the 3D scan from the tilting 2D laser’s data, removes spurious measurements and estimates surface normals for each point;
- *geometric mapping and functional reasoning*, which involves the detection and classification of principal planar regions;
- *refinement algorithms*, which apply processing functions on demand in order to augment the data, reconstruct the object models and compute descriptors to be used for identity resolution.

The result of this pipeline is a semantically annotated point cloud that contains information about floors, ceilings, walls, tables, doors, knobs, handles and point clusters of objects. This is passed to the dynamic object store, where it is registered together with timestamps and the observed point clusters.

Once a new measurement has been processed by this pipeline, the contents of the dynamic object store are updated on a per-table/per-cupboard basis. For instance, every table is matched against the list of previously detected tables based on its position, area of overlap and visibility. If a match is

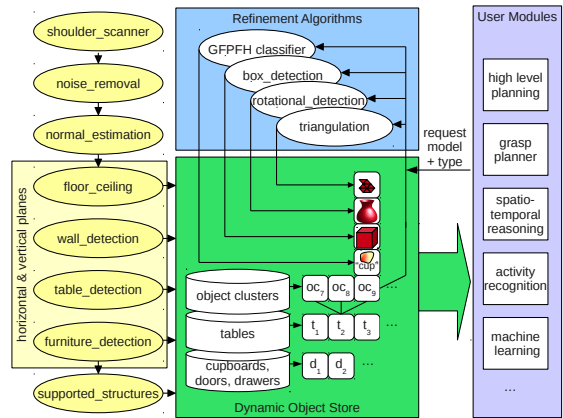


Fig. 3: Overview of the passive perception pipeline

found, the list of measurements for this table is extended with another *TableInstance*. The *TableInstance* structure is a tuple $\langle t, \mathcal{O} = \{oc_1, oc_2, \dots\} \rangle$, where t denotes the time at which the data was observed, and \mathcal{O} is a list of object clusters (i.e. subsets of the original point cloud). These object clusters oc_i are simply unstructured lists of points, with no semantics attached to them until they get processed and refined as described in the next section.

B. Object Refinement

Each observation of an object (clusters oc_i) is used to estimate, validate and update the geometric information stored about the object. Currently, the following specialized geometrical reconstruction mechanisms are combined in our system:

- *box reconstruction* fits a rectangular box as described in [17].
- *rotational reconstruction* approximates point clusters with a surface of revolution described by a symmetry axis and a contour line [18].
- *primitive shape reconstruction* can fit primitive geometric shapes such as cylinders, cones and spheres (cf. [19]).
- *triangulation* connects points to form a surface model without specific assumptions about the nature of the underlying geometry [20].

The selection of the best fitting model is detailed in [15].

Apart from these purely geometrical fitting methods, we also perform object categorization using a two-level classification scheme based on per-point fast point feature histograms (FPFHs) ([21]), which can be combined into a global fast point feature histogram (GFPFH) for each object. These features, in combination with machine learning techniques, allow us to recognize previously seen objects based on the geometric signature provided by GFPFHs [22].

These more descriptive models can now be used to compare clusters with each other in order to compute similarity probabilities used in the next section. These similarities are based on dimensions for the box models or radius and height for cylinders. For refinements like surfaces of revolution, where a feasible comparison method is not obvious, we can

employ a histogram intersection kernel to compare GPFH histograms for two observed clusters.

IV. OBJECT IDENTITY RESOLUTION

A very important subtask of our system is the task of object identity resolution. Whenever new observations arrive in the dynamic object store, we are interested in determining which of our current observations correspond to which past observations and which real-world entities are associated with them. Only by solving this problem can a robot create the notion of an *object*, i.e. a certain (not necessarily uniquely identifiable) entity in the real world.

Consider Figure 4, where we schematically depict a number of measurements over time (at times t_{k-2} , t_{k-1} and t_k) for three tables. On the tables, there are instances of two different object categories (represented as squares and circles). These may get replaced, moved, taken away or added by an agent not perceived by our system. The most interesting questions are: At any point in time, which *object* is each observation associated with, which objects have been moved where, and which objects have remained at the same position?

At time t_{k-1} , the robot changed its orientation and is therefore unable to observe tables 2 and 3. In order to be able to answer queries about objects that are temporarily out of view (depicted in grey in Figure 4), we need to maintain a sensible belief state which assumes that objects have not been moved unless we perceive evidence to the contrary (e.g. the object has appeared elsewhere).

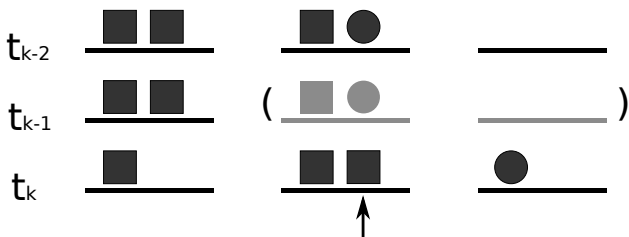


Fig. 4: Exemplary object identity resolution problem: Where was this square before?

We apply state-of-the-art statistical relational learning methods in order to tackle this problem. In statistical relational models [23], we can capture the complex interactions between objects that were observed, their attributes and relations – and represent these interactions generally, abstracting away from concrete entities. Most importantly, statistical relational models are capable of considering all the observations and facts that are relevant simultaneously, achieving a posterior belief on the associations between objects and observations that is guaranteed to be *globally consistent*.

A. Markov Logic Networks

In particular, we use Markov logic networks (MLNs) [24] as our representation formalism, for they combine the full power of first-order logic with the probabilistic semantics of graphical models. Formally, an MLN L is given by a set of pairs $\langle F_i, w_i \rangle$, where F_i is a formula in first-order logic

and w_i is a real number, the weight of formula F_i . For each finite domain of discourse D (set of constants), an MLN L defines a *ground Markov random field* $M_{L,D}$, whose set of variables X we obtain by grounding all the predicates in L with the entities in D , and whose set of weighted features is given by groundings of L 's formulas. $M_{L,D}$ specifies a probability distribution over the set of possible worlds, i.e. the set of possible assignments of truth values to each of the ground atoms in X , as follows,

$$P_w(X = x) = \frac{1}{Z} \exp \left(\sum_i w_i n_i(x) \right) \quad (1)$$

where $n_i(x)$ denotes the number of true groundings of F_i in x and Z is a normalization constant.

B. An MLN for Object Identity Resolution

Figure 5 shows the concrete MLN model we use to solve the problem of object identity resolution. As entity types, the model considers objects that can potentially be observed (*objects*), observations pertaining to these objects (*clusters*) as well as abstract entities representing time and basic object shapes. The time domain is heavily discretized; we differentiate only between the most recent point in time (*Now*) and points in time that preceded it (*Past*).

As soon as new observations come in, we can apply the model to update our beliefs. In our system, a new scan arrives every 10 seconds, which is therefore our time discretization step.

Markov logic networks are particularly well-suited for the representation of such a model, as they allow us to specify the various hard constraints that any valid entity-observation association must satisfy in first-order logic. At the same time, probabilistic rules (which either increase or decrease the likelihood of associations) can be expressed as soft constraints. Any beliefs computed by the model are guaranteed to be probabilistically sound and globally consistent with respect to the constraints that were specified.

Regarding our application, the key predicate we are interested in is the association between observations/clusters and objects (*is(cluster, object)*). We require that, in any possible world, every object can be associated with at most one cluster for any point in time (formula 2). Furthermore, similarity along any dimension should increase the degree to which we believe there to be an association (e.g. similarity with respect to spatial extents, *similarExt(cluster, object)*). Each similarity feature influences our beliefs about cluster-object associations (e.g. formula 12; other similarity features are handled equivalently and were omitted for brevity). Because the basic shape we detect in a cluster is a particularly discriminative feature, we consider shape explicitly (formula 11).

Apart from cluster-object associations, we need to be able to determine which clusters are equivalent (*same(cluster, cluster)*). By definition, two clusters are equivalent (in any possible world) if and only if there is some object with which both are associated (formula 4). Furthermore, since clusters have positions in space, we add a similarity feature over pairs of clusters (*similarPos(cluster, cluster)*, formula 13). As basic axioms, we require cluster equivalence to be reflexive

i	w_i	F_i	
1	(hard)	$\text{observed}(c,t1) \wedge \text{observed}(c,t2) \rightarrow (t1=t2).$	<i>domain declarations</i> $\text{domTime} = \{\text{Now}, \text{Past}\}$ $\text{domShape} = \{\text{Cylinder}, \text{Box}\}$ <hr/> <i>predicate declarations</i> $\text{observed}(\text{cluster}, \text{domTime})$ $\text{same}(\text{cluster}, \text{cluster})$ $\text{similarExt}(\text{cluster}, \text{object})$ $\text{similarPos}(\text{cluster}, \text{cluster})$ $\text{is}(\text{cluster}, \text{object!})$ $\text{shape}(\text{object}, \text{domShape!})$ $\text{cshape}(\text{cluster}, \text{domShape!})$ $\text{areaObserved}(\text{cluster})$ $\text{persists}(\text{cluster})$
2	(hard)	$\text{is}(c,o) \wedge \text{is}(c2,o) \wedge \text{observed}(c,t) \wedge \text{observed}(c2,t) \rightarrow (c=c2).$	
3	(hard)	$\text{is}(c,o) \wedge \text{is}(c2,o) \wedge \text{observed}(c,\text{Now}) \wedge \text{observed}(c2,\text{Past}) \wedge \text{persists}(c2) \rightarrow (c=c2).$	
4	(hard)	$\text{same}(c,c2) \leftrightarrow \exists o. (\text{is}(c,o) \wedge \text{is}(c2,o)).$	
5	(hard)	$\text{same}(a,a).$	
6	(hard)	$\text{same}(a,b) \rightarrow \text{same}(b,a).$	
7	(hard)	$\text{same}(x,a) \wedge \text{same}(y,a) \wedge \text{observed}(x,t) \wedge \text{observed}(y,t) \rightarrow (x=y).$	
8	(hard)	$\text{areaObserved}(c) \rightarrow \neg \text{persists}(c).$	
9	(hard)	$\text{is}(c,o) \wedge \text{is}(pc,o) \wedge \text{areaObserved}(c) \wedge \neg \text{areaObserved}(pc) \rightarrow \neg \text{persists}(pc).$	
10	$\log(0.90/0.10)$	$\text{persists}(c)$	
11	$\log(0.95/0.05)$	$\text{cshape}(c,cl) \wedge \text{shape}(o,cl) \rightarrow \text{is}(c,o)$	
12	$\log(0.83/0.17)$	$\text{similarExt}(c,o) \rightarrow \text{is}(c,o)$	
13	$\log(0.83/0.17)$	$\text{similarPos}(c,c2) \wedge \text{observed}(c,\text{Past}) \wedge \text{observed}(c2,\text{Now}) \rightarrow \text{same}(c,c2)$	

Fig. 5: Markov logic network for object identity resolution. Free variables in formulas are implicitly assumed to be universally quantified. The formulas specified as *hard* essentially have an infinitely large weight, which, in practice, is substituted by a sufficiently large real number. The domains *domTime* and *domShape* are fixed across all instantiations of the model, which is why we declare them explicitly. The predicate declarations indicate the domains to which the predicates are applicable. If, in such a declaration, an argument is suffixed by an exclamation mark, the predicate is declared as functional, i.e. $\text{pred}(a,b!)$ declares that for each entity of type a , there must be exactly one entity of type b for which the predicate is to hold in any possible world.

and symmetric (formulas 5 and 6), and we require that, for any point in time, there can be at most one cluster that is equivalent to any other cluster (formula 7).

Moreover, we need to cope with partial observability: The model needs to be able to handle the fact that areas in which we previously made observations can currently be unobserved ($\text{areaObserved}(\text{cluster})$) and maintain an appropriate belief about these areas (and the cluster-object associations therein). By default, we assume that the observations in unobserved areas are likely to carry over into the present ($\text{persists}(\text{cluster})$, formulas 8 and 10), unless there is evidence to the contrary (which is the case if another cluster whose area *is* currently observed refers to the same object, formula 9). With respect to the consistency of cluster-object associations, we treat persisting clusters as if they were currently observed (formula 3).

C. Application

We apply our MLN by instantiating it with the universe of objects that we will consider (providing *shape* for each object) and the set of relevant observations that were made, i.e. the set of clusters that are currently being observed and the most recent past clusters in all areas of the environment (including, in particular, the most recent clusters in areas that are currently unobserved, for which we need to determine whether or not we should believe they persist).

As evidence, we also need to provide the degree to which an observation/cluster matches the prototypical appearance of any of the objects under consideration, i.e. the degree to which our similarity features hold. For each similarity feature, we calculate a distance measure (e.g. the Mahalanobis distance in three dimensions for spatial extents) and map the (normalized) distances $d \in \mathbb{R}_0^+$ we obtain to similarity beliefs in $[0, 1]$ using $d \mapsto 2 - 2/(1 + \exp(-d))$, a function that decays approximately linearly at first and asymptotically approaches 0. In this way, we obtain a similarity belief for each cluster-object pair and each similarity feature. Each belief constitutes *soft evidence* (pertaining to ground atoms such as $\text{similarExt}(\text{Cluster}_i, \text{Object}_j)$).

We apply the model whenever we obtain new scans we want to interpret. Inference results from a previous iteration thus become soft evidence for the current iteration, i.e. results from the *Now* time step become evidence for the *Past* time step in the subsequent iteration. Specifically, we add as soft evidence the beliefs on cluster-object associations (i.e. *is* ground atoms). Moreover, in order to account for areas that are currently unobserved, we artificially inject evidence on the predicate *observed*: Whenever we previously inferred that an unobserved cluster persists (with probability p), we assume, in the next application of the model, that it is observed (with degree of belief p).

To perform inference in the light of both soft and hard evidence, we apply a specifically adapted Monte Carlo method for soft evidential update [25]. One inference run can essentially be viewed as a complete filtering step (combined prediction and update).

V. EXPERIMENTAL RESULTS

Since most of the geometric processing components are drawn from previous work, we focus on object identity resolution in our evaluation. In the following, we present two instances of the example scenario shown earlier (see Fig. 4) in order to illustrate that our approach is generally sound. As ground truth, assume that in the first time step, none of the objects moved but the robot was repositioned such that it could only perceive table 1; and in the second time step, the second box (square) from table 1 was moved to table 2 and the plate (circle) was moved to table 3.

Figure 6a depicts the results for a case where the initial association of clusters to objects is known (e.g. the first box is the object with identifier *Box1*, etc.) and the universe of objects comprises three boxes and one plate (all of which were observed in the first step). The image indicates that both the association of clusters with objects and the equivalence of pairs of clusters over time is clearly consistent with common sense (and the modes of the respective distributions indeed correspond to ground truth). Our positional similarity beliefs were high (0.95) for clusters that are drawn at the

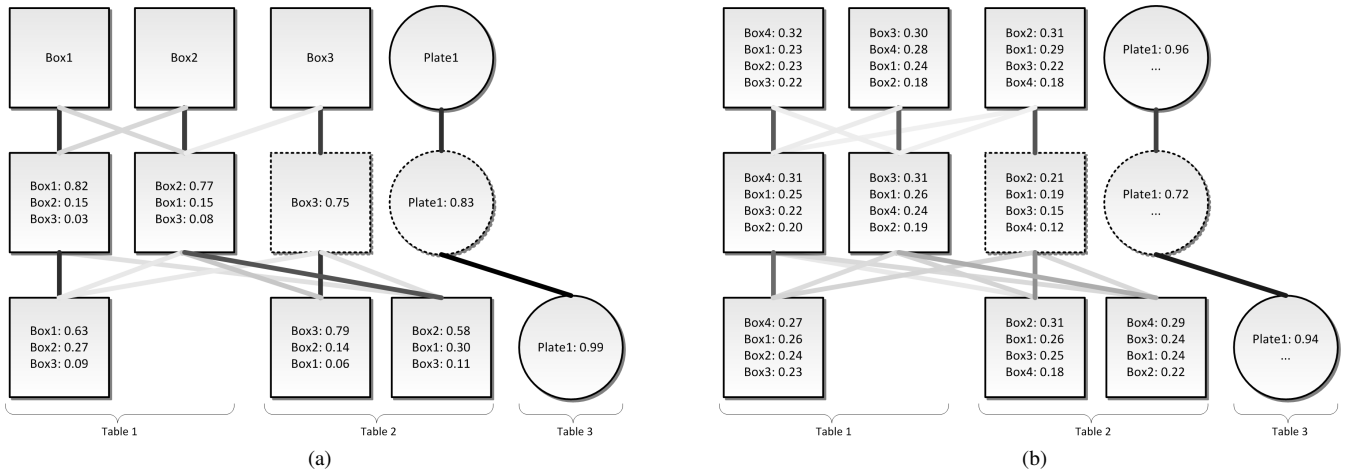


Fig. 6: Illustration of inference results. Every square/circle represents an observation that was made (cluster). Dotted squares/circles are in fact not observed but are shown in order to illustrate beliefs on unobserved areas (observation persistence). Cluster-object associations are listed in cluster labels, cluster equivalence is reflected in the blackness of lines connecting clusters (black = 1.0, white = 0.0). In case (a), the universe comprises three boxes and one plate, and the cluster-object associations for time step 1 are known. In case (b), the universe comprises four boxes and one plate, and we have no evidence on initial associations.

same horizontal coordinate in Figure 6a, and appropriately low otherwise (0.05). The extents for all objects having the same shape were considered as equally similar. Note that we purposely chose to assign a fairly low weight to formulas indicating similarity (e.g. formula 13) in order to amplify the decay of beliefs over time, facilitating visualization of low probabilities. Beliefs thus drop substantially in a single time step. Higher weights should be chosen for real-world applications. (Given enough data, it would be most sensible to obtain them using machine learning techniques.)

In Figure 6b, we did not assume prior knowledge on cluster associations and added an additional box to the universe of objects (i.e. all boxes are never observed simultaneously). We observe that, even though beliefs on associations are essentially uniform (discrepancies are due only to approximate inference), cluster equivalence (tracking) is still inferred correctly.

VI. CONCLUSIONS AND FUTURE WORK

We presented a complex framework that deals with many of the harder problems found in current 3D robotics perception systems. We showed how to integrate the individual perceptual and reasoning modules into a powerful tool that can enable a robot to detect a wide variety of objects of daily use in human living environments and keep track of cluster-object associations over time based on a probabilistic first-order model that considers all the relevant aspects collectively in order to achieve globally consistent beliefs even in the presence of ambiguity and partial observability. We achieve this by expanding on previously published methods for geometric processing of 3D data, integrating them into a common framework, and using probabilistic methods to interpret the observations in a sensor-independent manner.

However, some topics will need to be addressed in future work. Although the probabilistic method we proposed is,

from a theoretical point of view, the method of choice (for it possesses precisely the required degree of expressiveness), it does not scale arbitrarily. If real-time performance is to be achieved, it may be advisable to explore methods that make simplifying assumptions, trading off soundness and speed. Furthermore, we will need to find ways of handling vast amounts of data consistently, so as to remember important aspects while discarding redundant information. Moreover, the investigation of systems that make use of the information gathered by the methods we proposed warrants attention. These include, for instance, high-level planning layers and activity recognition systems, where a detection of a sequence of objects involved is key.

On the perception side, a topic that may require further investigation is the addition of further refinement algorithms, as well as the inclusion of different sensing modalities like camera images or disparity images from stereo vision. As a consequence of having more of these methods and data representations available, it may become necessary to (adaptively) learn which method is likely to produce the best results for a specific observation.

REFERENCES

- [1] R. C. Nelson and I. A. Green, "Tracking objects using recognition," *Pattern Recognition, International Conference on*, vol. 2, p. 21025, 2002.
- [2] G. Andrienko, N. Andrienko, and S. Wrobel, "Visual analytics tools for analysis of movement data," *ACM SIGKDD Explorations*, vol. 9, no. 2, pp. 38–46, 2007.
- [3] L. Liao, D. Fox, and H. Kautz, "Extracting places and activities from gps traces using hierarchical conditional random fields," *International Journal of Robotics Research*, 2007.
- [4] N. Landwehr, B. Gutmann, I. Thon, M. Philipose, and L. D. Raedt, "Relational transformation-based tagging for human activity recognition," in *Proceedings of the 6th Workshop on Multi-Relational Data Mining (MRDM)*, Warsaw, Poland, 2007.

- [5] S. Intille, K. Larson, E. M. Tapia, J. Beaudin, P. Kaushik, J. Nawyn, and R. Rockinson, "Using a live-in laboratory for ubiquitous computing research," in *Proceedings of PERVASIVE 2006*, K. Fishkin, B. Schiele, P. Nixon, and A. Quigley, Eds., vol. LNCS 3968. Berlin Heidelberg: Springer-Verlag, 2006.
- [6] I. P. Fellegi and A. B. Sunter, "Ad/dx.doi.org/10.2307/2286061," *Journal of the American Statistical Association*, vol. 64, no. 328, pp. 1183–1210, 1969.
- [7] P. Singla and P. Domingos, "Entity Resolution with Markov Logic," in *In ICDM*. IEEE Computer Society Press, 2006, pp. 572–582.
- [8] I. J. Cox and S. L. Hingorani, "An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 2, pp. 138–150, February 1996.
- [9] T. Schmitt, R. Hanek, M. Beetz, S. Buck, and B. Radig, "Cooperative probabilistic state estimation for vision-based autonomous mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, October 2002.
- [10] D. Schulz, W. Burgard, D. Fox, and A. B. Cremers, "Tracking Multiple Moving Targets with a Mobile Robot using Particle Filters and Statistical Data Association," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2001, pp. 1665–1670.
- [11] A. Shirazi and E. Amir, "First-order logical filtering," in *IJCAI'05: Proceedings of the 19th international joint conference on Artificial intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2005, pp. 589–595.
- [12] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [13] V. Lepetit and P. Fua, "Keypoint recognition using randomized trees," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 9, pp. 1465–1479, Sept. 2006.
- [14] R. B. Rusu, Z. C. Marton, N. Blodow, A. Holzbach, and M. Beetz, "Model-based and Learned Semantic Object Labeling in 3D Point Cloud Maps of Kitchen Environments," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, St. Louis, MO, USA, October 11-15 2009.
- [15] Z.-C. Marton, D. Pangercic, N. Blodow, J. Kleinhellefort, and M. Beetz, "General 3D Modelling of Novel Objects from a Single View," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, October 18-22 2010.
- [16] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz, "Close-range Scene Segmentation and Reconstruction of 3D Point Cloud Maps for Mobile Manipulation in Human Environments," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, St. Louis, MO, USA, October 11-15 2009.
- [17] Z. C. Marton, L. Goron, R. B. Rusu, and M. Beetz, "Reconstruction and Verification of 3D Object Models for Grasping," in *Proceedings of the 14th International Symposium on Robotics Research (ISRRO9)*, Lucerne, Switzerland, August 31 - September 3, 2009.
- [18] N. Blodow, R. B. Rusu, Z. C. Marton, and M. Beetz, "Partial View Modeling and Validation in 3D Laser Scans for Grasping," in *9th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, Paris, France, December 7-10 2009.
- [19] R. Schnabel, R. Wahl, and R. Klein, "Efficient RANSAC for Point-Cloud Shape Detection," *Computer Graphics Forum*, vol. 26, no. 2, pp. 214–226, June 2007.
- [20] 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, May 12-17, 2009.
- [21] R. B. Rusu, N. Blodow, and M. Beetz, "Fast Point Feature Histograms (FPFH) for 3D Registration," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, May 12-17, 2009.
- [22] R. B. Rusu, A. Holzbach, G. Bradski, and M. Beetz, "Detecting and segmenting objects for mobile manipulation," in *Proceedings of IEEE Workshop on Search in 3D and Video (S3DV), held in conjunction with the 12th IEEE International Conference on Computer Vision (ICCV)*, Kyoto, Japan, September 27 2009.
- [23] L. Getoor and B. Taskar, *Introduction to Statistical Relational Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2007.
- [24] M. Richardson and P. Domingos, "Markov Logic Networks," *Mach. Learn.*, vol. 62, no. 1-2, pp. 107–136, 2006.
- [25] D. Jain and M. Beetz, "Soft Evidential Update via Markov Chain Monte Carlo Inference," in *33rd Annual German Conference on Artificial Intelligence (KI 2010)*, 2010.