The Dynamic Anchoring Agent: A Probabilistic Object Anchoring Framework for Semantic World Modeling

Zongyao Yi, Martin Günther
German Research Center for AI (DFKI)
Osnabrück, Germany
{zongyao.yi,martin.guenther}@dfki.de

Joachim Hertzberg
Osnabrück University and
German Research Center for AI (DFKI)
Osnabrück, Germany
joachim.hertzberg@uos.de

Abstract

Semantic world modeling has been studied extensively, with the goal of enabling robots to understand and interact with their environment. However, existing approaches to semantic world modeling rely on well-defined perceptual data, such as distinct visual features. In situations where different objects are difficult to distinguish based on perceptual data alone, the resulting world model will be ambiguous and inconsistent. To address this challenge, we present the Dynamic Anchoring Agent (DAA), a probabilistic object anchoring framework for semantic world modeling that uses domain knowledge and reasoning to handle the ambiguity of sensor data through probabilistic anchoring. It includes a Multiple Hypothesis Tracker (MHT) as a filter for noisy observations, and a knowledge base that encodes domain knowledge and scene context to reduce uncertainty in the anchoring process. The framework is evaluated on both synthetic and real-world datasets, demonstrating its effectiveness in resolving association ambiguities in the presence of identical-looking instances. It has also been integrated into a real robot platform. We show that with the help of domain knowledge and scene context, the proposed framework outperforms traditional pure data-based algorithms in terms of identification accuracy, and can effectively resolve ambiguities between sensor-identical object instances.

Introduction

In the context of robot manipulation, task planners generate plans concerning symbolic entities, while the robot executes the plans on the sensorimotor level. Consider an agent in a household environment. Often, the robot executes actions in different locations. While the robot is operating in the kitchen, the object states in the living room may have changed. A prerequisite for successful task execution is the correct correspondence between the entities referred to by the task planner and the sensory perception of the agent. Maintaining this correspondence over time is called the anchoring problem (Loutfi, Coradeschi, and Saffiotti 2005). In our example, when the household robot comes back to the living room, it has to figure out which existing symbols correspond to which perceived objects in the changed scene through the use of anchoring.

The main challenges in the anchoring problem are twofold. First, the agent's sensory observations are noisy and partial, especially with object occlusion and dynamics. Second, association ambiguities arise from dynamic object instances with identical perceptual features, e.g., object appearance and position. Since most existing methods approach the anchoring problem by relying on the distinct features of the individual objects, the association ambiguities cannot be effectively resolved when facing these identical-looking objects. In this paper, we propose to resolve ambiguities with the help of domain knowledge and logical reasoning. Specifically, we use the Multiple Hypothesis Tracking algorithm as a filter for the noisy observations, and on top of that, we maintain a scene graph and exploit the encoded scene context to reduce the uncertainty in the anchoring process. In summary, we make the following contributions:

1. We present the open-source probabilistic object anchoring framework DAA[1] for semantic world modeling, which explicitly solves the data association problem using domain knowledge and logical reasoning.
2. We compare the identification performance of the proposed framework against the MHT algorithm in terms of ID F1 score, precision, and recall.
3. We demonstrate the effectiveness of the framework in resolving association ambiguities in the presence of identically looking instances.

Related Work

In (Loutfi, Coradeschi, and Saffiotti 2005), the anchoring problem was presented and the elements and structure of anchoring frameworks were defined, including a symbolic system, a perceptual system, and an anchoring module connecting the two. They also integrated multimodal perceptual information. *Find*, *reacquire* and *track* functionalities are introduced so that traditional bottom-up anchoring approaches work in conjunction with top-down approaches to information acquisition. In their subsequent work (Loutfi et al. 2008), more sophisticated knowledge representation and reasoning are incorporated to enrich the anchoring process. Daoutis et al. extend it to multi-agent settings in (Daoutis, Coradeschi, and Loutfi 2012). More recently, Persson et

[1] https://github.com/copda/copda

al. employ machine learning and dynamic distributional clauses in their anchoring framework (Persson et al. 2020; Zuidberg Dos Martires et al. 2020). In (Zuidberg Dos Martires et al. 2020), rules are learned instead of being hand-coded. The object-percept association is handled by a match function, which is a support vector machine trained on human-labeled data. The match function computes the similarities of object attributes, mostly the similarity of positions. In all of the above approaches, object-to-percept associations are made based on the perceptual similarities of individual objects. The uncertainty about associations itself is not explicitly handled.

Another line of work includes (Elfring et al. 2013; Wong 2017; Wong, Kaelbling, and Lozano-Pérez 2015), where data association is explicitly handled in a probabilistic manner. (Elfring et al. 2013) is a representative of tracking-based methods. They use MHT to maintain all possible object states up to N frames. Although perceptions are grounded in conceptual space, the implied semantic information is not exploited in the association process, and the potential association errors made by the MHT are not addressed. On the other hand, Wong (2017) approaches the anchoring problem using clustering under the assumption that the world is semi-static. Same as (Elfring et al. 2013), it also relies on probabilistic inference at the perceptual level without logical reasoning at the semantic level.

Approaches that use scene context in the form of pairwise spatial relations between objects to address the anchoring problem can be found in (Ruiz-Sarmiento et al. 2017) and (Günther et al. 2018). They use a Conditional Random Field (CRF) to combine the results of a low-level object recognition system with a trained matching function based on geometric features and scene context. In our work, we go one step further by employing Multiple Hypothesis Tracking. MHT delays the tracking decision for several more frames, thereby improving the tracking results because it has access to more information from several frames before committing to the most likely hypothesis. We also add a knowledge layer that allows the user to use domain knowledge in the anchoring process.

Preliminaries

Problem Definition

Consider a scene containing an unknown number of objects. The number at time t, denoted by N_t, changes as objects enter or leave the scene. Objects are assumed to have static and dynamic attributes, denoted by s_t and x_t, respectively. A black-box perceptual module provides direct estimates for these attributes. Let θ_t^k be the kth object's attributes of interest at time t, then $\theta_t^k = (s_t^k, x_t^k)$ with $k \in \{1, \ldots, N_t\}$. At each time step, the agent has a limited view of the scene, and the perceptual module provides M_t observations of the object attributes within the view, denoted as $O_t = \{o_t^m \mid \text{for } m = 1, \ldots, M_t\}$. We assume that the perceptual module has a detection rate of p_{DT} and a false detection rate of p_{FT}. In general, we try to recover the object states $\Theta_{1:t} = \{\Theta_1, \ldots, \Theta_t\}$ from the observations in an online fashion, namely the posterior $p(\Theta_{1:t} \mid O_{1:t})$, where $\Theta_t = \{\theta_t^k \mid \text{for } k = 1, \ldots, N_t\}$.

Data Association

Without the object-to-observation association, the target distribution $p(\Theta_{1:t}|O_{1:t})$ is difficult to compute. There are several approaches to the data association problem, such as Multiple Hypothesis Tracking (Reid 1979) or Markov chain Monte Carlo data association (Oh, Russell, and Sastry 2009). Here we use the MHT approach as in (Kim et al. 2015) and (Elfring et al. 2013). Specifically, an association variable $\boldsymbol{z}_t$ is introduced, which is a vector of length M_t, with each component z_t^m being the value of the associated object index k, or 0 for false detections. The MHT maintains the full distribution of $\boldsymbol{z}_t$, and the posterior becomes the expectation of $p(\Theta_{1:t}|\boldsymbol{z}_{1:t}, O_{1:t})$ under $p(\boldsymbol{z}_{1:t}|O_{1:t})$. The joint distribution $p(\Theta_{1:t}, \boldsymbol{z}_{1:t}|O_{1:t})$ is maintained in a tree structure, with nodes representing association hypotheses. Since the full joint distribution is impossible to maintain due to the exponentially growing tree size, approximation techniques such as N-scan pruning are used. However, these pruning schemes introduce irreversible association errors. Furthermore, MHT implicitly maximizes the likelihood $p(O_{t-N:t}|\boldsymbol{z}_{t-N:t}, \Theta_{t-N:t})$ wrt. the association variable $\boldsymbol{z}_{t-N:t}$, and in some cases the likelihood is not unimodal, for example, when two visually identical objects swap positions. In these cases, using the perceptual information alone is not sufficient to resolve the ambiguities. To address these issues, we introduce the symbolic system; on top of the results from MHT, the anchoring process solves the data association problem with the inferred scene context.

Symbolic System

To compensate for the drawbacks of MHT, we exploit the latent semantics in data and domain knowledge. In particular, we introduce the symbolic system, which contains a set of symbols $\Pi = \{\pi_k\}$ representing physical objects, a set of predicates $\Phi = \{\phi_j(\pi_k, value)\}$ describing the attributes of these objects, and a set of relational predicates $R = \{r_l(\pi_k, \pi_v)\}$ that encode the spatial relationships between objects. In summary, the symbolic system is described as the union of the three: $S = \Pi \cup \Phi \cup R$. With the spatial relationships, we can augment the log-likelihood with a relational likelihood term: $\log p(R_t, O_t|\boldsymbol{z}_t, \Theta_t) = \log p(R_t|O_t, \boldsymbol{z}_t, \Theta_t) + \sum_{m=1}^{M_t} \log p(o_t^m|\boldsymbol{z}_t, \Theta_t)$.

Anchoring Process

Let T_i be the ith track from the MHT defined from time u to v, and $T_i = \{o_{u:v}^m \mid z_{u:v}^m = i\}$. The anchoring process then assigns these tracks to symbols by maximizing the augmented log-likelihood mentioned above. We define an anchor as a tuple of a symbol defined in the symbolic system and the corresponding track, namely $\alpha_k = (\pi_k, T_i)$. The anchoring results are represented as a set of anchors. Thus, the state of the object instance π_k between time u and v can be computed as $p(\theta_{u:v}^i|o_{u:v}^m, z_{u:v}^m = i)$.

Dynamic Anchoring Agent

This section describes the Dynamic Anchoring Agent (DAA) in detail. The DAA is structured as shown in Figure 1 and consists of three components: Multiple Hypothesis Tracker, Anchor Management, and Knowledge Base. The DAA operates in a bottom-up fashion and is driven by perceptual data from a perception module. The perceptual data can be of different modalities, such as position, orientation, object class, and color distribution. The Multiple Hypothesis Tracker (MHT) serves as an initial filter to handle false detections. As a result, the perceptual data is structured as short, continuous tracks of percepts. Next, the Anchor Management (AM) module performs the anchoring process, in which the track-symbol assignments are resolved and stored as a set of anchors. Finally, the Knowledge Base (KB) updates the semantic information about object instances based on the latest anchors and supports the AM in the next anchoring process. The KB also provides interfaces to high-level modules, such as the task planner, for querying object states.

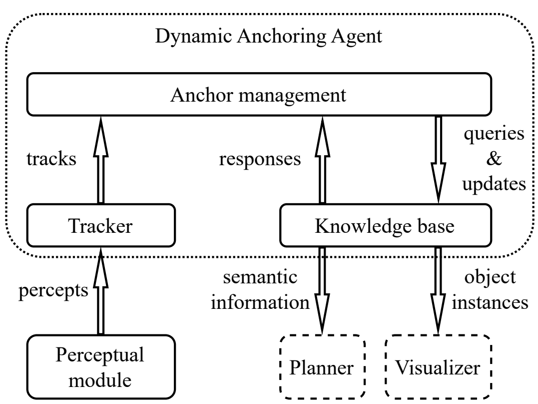


Figure 1: Framework structure of the DAA.

Multiple Hypothesis Tracker

Since the incoming perceptual data contains misdetections and false alarms, we implemented the Multiple Hypothesis Tracker (Reid 1979; Kim et al. 2015; Elfring et al. 2013) to handle the initial data association. The tracker maintains a set of multi-hypothesis trees, each of which contains possible sequences of percepts that originate from a real-world entity. In our demo domain, the percepts consist of three different object properties, namely position, object class, and color distribution; however, which properties are used is flexible and depends on the domain. Association scores are calculated as the weighted sum of the log-likelihoods of these properties. To limit the computational complexity, poorly scored branches are pruned and the N-scan pruning strategy is applied. However, according to the investigation in (Wong 2017), the pruning techniques of MHT come with some non-trivial limitations:

- Correct associations may be erroneously eliminated by the pruning process.
- Although the association decision is delayed by the N-scan window, there is no guarantee that the ambiguities will be resolved after N scans, and hence the algorithm will still commit to the erroneous associations.

Also, since the MHT only performs tracking and not anchoring, it cannot re-associate an object that reenters the scene after a long absence to its previous track (i.e., after tracking is lost). To account for these problems, the resulting tracks are assigned to temporary targets whose identities are decided during the anchoring process.

Anchor Management

Anchor management is the interface between the tracker and the knowledge base. It decides whether a track from the MHT should be associated with an existing symbol or create a new one by solving an optimal assignment problem. The key to the anchoring process is a matching function that evaluates proposed assignments based on perceptual data as well as scene context (in our case, spatial relations).

At each time step, the AM receives a set of tracks from the MHT, which can be divided into three groups: tracks associated with existing anchors, terminated tracks, and new tracks. The most recent percepts retrieved from the associated tracks are stored for the following process. For terminated tracks, the corresponding anchors are marked as lost. The optimal assignment between lost anchors and new tracks is then solved as follows.

Optimal Assignment First, we propose possible assignments by enumerating the combinations between dangling tracks and lost anchors. For each track, a new candidate anchor is also created to account for a possible new object. Each assignment is validated with the match function depending on static object attributes such as object types and colors. We then construct a graph where each node represents a valid assignment. By assuming that a track can only be assigned to a maximum of one anchor and vice versa, edges can be created between compatible assignments. The score of a compatible assignment set is computed as the weighted sum of sensory similarities and the scene graph distance given the expected spatial relations:

$$-\omega_o[\sum_{m:z_t^m \neq 0}^{M_t} \|o_t^m - \theta_t^{z_t^m}\|_2 + \sum_{m:z_t^m = 0}^{M_t} l(o_t^m)] - \omega_d d_{\boldsymbol{z}_t}$$

where $l(o_t^m)$ is a function that evaluates the possibility of the observation coming from a new instance based on prior knowledge, e.g. object counts and default locations. These quantities are calculated by a match function with the support of the knowledge base. The maximum score clique of the constructed graph contains the most likely assignments. For assignments connecting tracks to new anchors, new symbols are acquired by querying the knowledge base and used to initialize the new anchors. For the assignments with lost anchors, the terminated tracks are stored and replaced by the new tracks.

Finally, the relevant instances in the knowledge base can be updated with the latest percepts of the associated tracks.

Knowledge Base

The Knowledge Base (KB) module contains a database and a manager to access it. Together with the Prolog scripts that encode domain knowledge, it forms the symbolic system and provides query and update services to other modules such as the anchor management and task planner.

Knowledge Representation In the database, knowledge is represented by a logical component and a physical component. The logical component describes domain-specific concepts at different levels of abstraction and their relations. This domain-specific knowledge must be tailored to the specifics of the particular domain in which the system is operating and remains static during operation. In our implementation, we developed a small-scale ontology as shown in Figure 2. The physical component is the instantiation of the logical concepts and reflects the current semantic world states of the agent. The knowledge is encoded in the form of *(Subject, Predicate, Object)* triples and stored in an SQLite database.

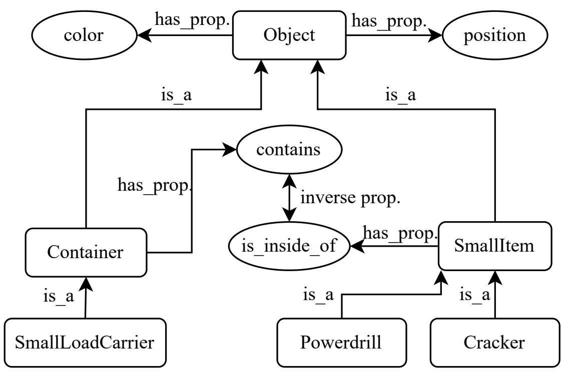


Figure 2: The small scale ontology

Knowledge Base Manager While the database serves as storage for the knowledge representation, the knowledge base manager provides logical reasoning capabilities. The reasoning capabilities are empowered by rules programmed in the Prolog language and can be divided into two groups: indirect relation inference (for properties and classes) and spatial relation inference.

Furthermore, the KB manager provides query and update services for other modules to access the knowledge base. The query service supports the anchoring process and provides semantic information to modules such as the task planner. It accepts the following queries:

- `is_potential_match`: Checks whether the given percepts potentially originate from the specified instance.
- `get_relation_graph`: Returns the spatial relations of the current world model.
- `infer_relation_graph`: Returns the resulting spatial relations given the hypothesized object states.
- `get_all_instances`: Returns all object instances along with their state estimates.

On the other hand, the update service enables other modules to update the knowledge base, including creating, deleting, and updating instances.

Consistency Checks Due to the potential errors from the anchoring process and MHT, the resulting updates to the world model may be inconsistent. We try to mitigate this by performing a consistency check before committing the updates. First, we check for conflicting spatial relations. For example, if `(container_1,contains,item_1)` is inferred by the knowledge base based on new sensor data, then the existing fact `(item_1,is_inside_of,container_2)` that was true previously is deleted. We also check for duplicate instances. For example, if `container_1` from the above example is lost and `container_2` is active, then `container_1` and `container_2` likely represent the same physical entity. In this case, we merge `container_1` with `container_2`.

Experiments

The experiments are divided into three parts. First, we evaluate the DAA with a 2D toy dataset and compare the performance with the original MHT. In the second part, we further demonstrate its effectiveness in resolving association ambiguities in various challenging situations. Finally, we show a qualitative example of the integration of DAA on a real robot.

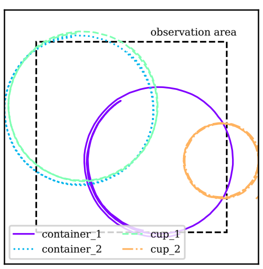
Evaluation on the Synthetic Dataset We created a synthetic dataset that contains sequences of object type and pose observations in chronological order $O = \{\{o_1^1, o_1^2, \dots\}, \{o_2^1, o_2^2, \dots\}, \dots\}$, and each observation is labeled with the ground truth object identifier. The dataset is generated as follows.

First, we define two types of objects: `container` and `cup`. To ensure that the algorithm can exploit the spatial relations, we set the number of objects of both types to 2, with one cup always inside a container. This allows us to reuse the predefined ontology shown in Figure 2, with `container` as a subclass of `Container` and `cup` as a subclass of `SmallItem`.

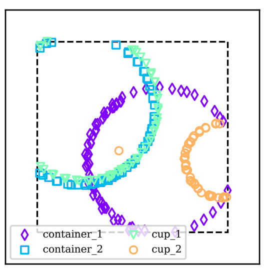
Objects are scattered and constrained in a 4×4 square space, and objects cannot overlap. To simulate a dynamic scene, objects follow random circular trajectories with random accelerations at each time step (Figure 3a). An object emits an observation with probability p_{DT} (Figure 3b). To simulate objects entering and disappearing from the scene, we define an observation area of size 3×3, centered in the scene area. Only the objects inside the observation area can be observed. The position observations are corrupted with zero mean Gaussian noise. With a probability of p_{FT}, a spurious observation is generated by sampling uniformly over the entire space. Each generated observation is labeled with the corresponding object identifier. Figure 3a shows one set of generated ground truth trajectories, where `container_2` contains `cup_1` and the other objects move independently. Figure 3b shows the rendered noisy observations that are fed into the MHT.

We sample trajectories with incremental step sizes ranging from 500 to 2000, with p_{DT} fixed at 0.6 and p_{FT} fixed at 0.05. For each step size, five sequences are sampled. For quantitative evaluation, we run our framework with the generated sequences and adopt identification accuracy / F1 score (IDF1), identification precision (IDP) and identification recall (IDR) as performance metrics (Ristani et al. 2016). Since the MHT is part of our framework, the results of both the MHT and the DAA can be recorded simultaneously during the runs. We compare the corresponding results with the Python package py-motmetrics[2]. The full comparison is listed in the Table 1.

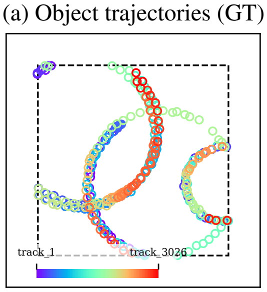
The generated dataset is challenging for the MHT because the objects exit and enter the observation area multiple times and from very different positions; and objects of the same type sometimes come very close in space (see Figure 3). This becomes more apparent as the trajectories become longer. The IDF1 trends of both MHT and DAA with increasing trajectory length is illustrated in Figure 4. The IDF1 scores of the MHT decrease significantly with increasing trajectory length, while those of the DAA decrease only slightly. From Table 1 we can see that the DAA also outperforms the MHT in terms of IDF1, IDP and IDR. A more intuitive comparison is shown in Figure 3c and Figure 3d, where we can see that the DAA provides a much more consistent labeling of the observations.



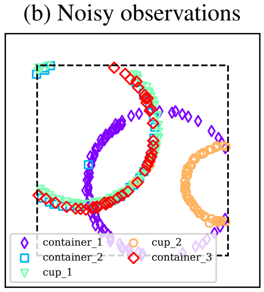
(a) Object trajectories (GT)



(b) Noisy observations



(c) MHT results



(d) DAA results

Figure 3: Visualization of the generated trajectories of length 2000 and the corresponding tracking and anchoring results. Data points have been downsampled for better visualization.

Experiment with Real Sensor Data In this section, we demonstrate with real camera data that our framework can distinguish visually identical instances based on scene context. To this end, we prepare 2 identical small load carrier (KLT) boxes, one containing a power drill and the other some colored boards. For perception, we use the deep 3D pose estimation system DOPE (Tremblay et al. 2018) together with an Asus Xtion Pro RGB-D camera. DOPE has been trained beforehand with the 3D models of the objects present in this experiment, including KLT and power drill. Besides pose estimation, the perception module is also integrated with a color classification component that takes the segmented color images as input and outputs color histograms of the individual object instances. Thus, the inputs to the DAA include object types, 3D positions, and color histograms. In this experiment, we also use the `contains/is_inside_of` relation pair, and the `contains/2` rule is defined as follows:

```
contains(Container, Item):-
  is_a(Container, onto:'Container'),
  is_a(Item, onto:'SmallItem'),
  close_enough(Container, Item),
  have_common_colors(Container, Item).
```

`close_enough` and `have_common_colors` are two predefined predicates. The former checks whether two objects are close in both space and time and the latter asserts if two instances have common colors. Note that although object color information is used while inferring `contains/is_inside_of` relation, it is not strictly necessary. We can, for instance, replace the `have_common_colors` with another rule that checks the

[2]https://github.com/cheind/py-motmetrics

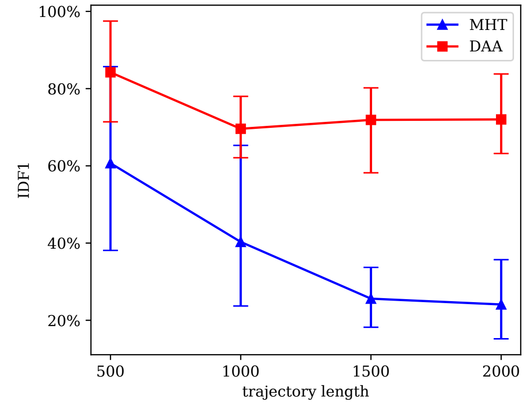


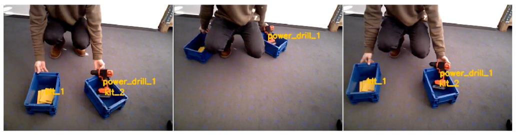
Figure 4: IDF1 scores of MHT and DAA with increasing trajectory length

Seq. length	IDF1		IDP		IDR	
	MHT	DAA	MHT	DAA	MHT	DAA
500	60.68	84.22	62.67	86.82	58.78	81.82
1000	40.30	69.60	41.94	72.28	38.80	67.10
1500	25.60	71.90	26.72	74.58	24.58	69.42
2000	24.10	72.02	25.06	74.66	23.24	69.54
Overall	37.67	**74.44**	39.11	**77.09**	36.35	**71.97**

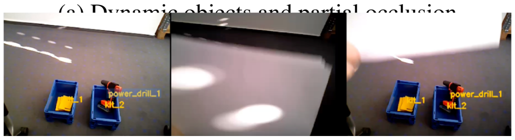
Table 1: Performance of the MHT and DAA on the identification metrics

correlation between object positions.

In the first run, both KLT boxes were moved behind the obstacle and then moved back to the center (see Figure 5a). For the second run, we completely occluded the camera for a period of time, causing all objects to be reported as lost. The objects were then revealed and detected again (see Figure 5b). In the third run (Figure 5c), we exchanged the positions of both boxes while the camera was occluded, and then revealed the objects again. As shown in Figure 5, both boxes were correctly identified in all runs, despite occlusion and object dynamics. The supplementary video can be viewed on our project page[3].



(a) Dynamic objects and partial occlusion



(b) Complete occlusion

(c) Complete occlusion and dynamic objects

Figure 5: Example results with live sensor data

Integration with Real Robot Platform As a final example, we show the integration of the DAA with the Mobipick robot platform (see Figure 6). The DAA is fully integrated with ROS and produces a semantic world model from sensor data that can be used by the robot for task planning and execution (see (Lima et al. 2023) for details).

Conclusion and Future Work

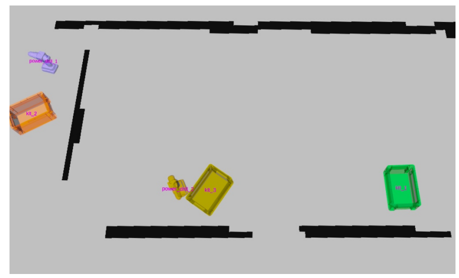
In this paper, we present a probabilistic object anchoring framework for semantic world modeling, which is available on GitHub.[4] The proposed framework combines the probabilistic tracker (MHT) with domain knowledge and logical reasoning. As demonstrated in the experiments, by exploiting the `contains/is_inside_of` relations alone, our framework can effectively resolve association ambiguities arising from the presence of visually identical objects. The quantitative comparison between our framework and the MHT-only tracker shows that our framework outperforms MHT by almost 50% in terms of IDF1 score.

On the other hand, we are also aware of some limitations of our framework. In the experiments, only one pair of spatial relations is considered. This quickly becomes insufficient when there are multiple instances with the same relations, for example, when both KLT boxes in the second experiment contain power drills. In the future, other relations such as causal or temporal relations (Hafri and Firestone 2021) can be added to handle the ambiguities involved. We also note that although some association errors from the MHT can be revised during the anchoring process, this revision itself can still be error-prone and is irreversible. This is due to the fact that the DAA is a MAP approximation to the real association distribution, which can be multimodal. A possible solution would be representing object attributes with particle filters combined with batch algorithms such as the MCMCDA (Oh, Russell, and Sastry 2009). Finally, it is also beneficial to fuse 3D sensor data, such as depth maps from 3D cameras, as they provide additional information about free and occupied space. This information can greatly reduce the uncertainty of associations. The fusion of object and metric spatial information can be found in (Wong 2017).

[3] `https://sites.google.com/view/dynamicanchoringagent`

[4] `https://github.com/copda/copda`

(a) The Mobipick robot in the demo environment (image reproduced with permission from (Lima et al. 2023)).



(b) The resulting semantic world model at a later point in time (after some objects were moved).

Figure 6: Demonstration on real robot

Acknowledgments

This work is supported by the CoPDA and ExPrIS projects through grants from the German Federal Ministry of Education and Research (BMBF) with Grant Numbers 01IW19003 and 01IW23001. The DFKI Niedersachsen (DFKI NI) is sponsored by the Ministry of Science and Culture of Lower Saxony and the VolkswagenStiftung.

References

Daoutis, M.; Coradeschi, S.; and Loutfi, A. 2012. Cooperative knowledge based perceptual anchoring. *Int. J. Artif. Intell. Tools* 21(3).

Elfring, J.; van den Dries, S.; van de Molengraft, M. J. G.; and Steinbuch, M. 2013. Semantic world modeling using probabilistic multiple hypothesis anchoring. *Robot. Auton. Syst.* 61(2):95–105.

Günther, M.; Ruiz-Sarmiento, J. R.; Galindo, C.; González-Jiménez, J.; and Hertzberg, J. 2018. Context-aware 3D object anchoring for mobile robots. *Robotics Auton. Syst.* 110:12–32.

Hafri, A., and Firestone, C. 2021. The perception of relations. *Trends Cogn. Sci.* 25(6):475–492.

Kim, C.; Li, F.; Ciptadi, A.; and Rehg, J. M. 2015. Multiple hypothesis tracking revisited. In *2015 IEEE International Conference on Computer Vision, ICCV 2015*, 4696–4704.

Lima, O.; Günther, M.; Sung, A.; Stock, S.; Vinci, M.; Smith, A.; Krause, J. C.; and Hertzberg, J. 2023. A physics-based simulated robotics testbed for planning and acting research. In *ICAPS Workshop on Planning and Robotics (PlanRob 2023)*.

Loutfi, A.; Coradeschi, S.; Daoutis, M.; and Melchert, J. 2008. Using knowledge representation for perceptual anchoring in a robotic system. *Int. J. Artif. Intell. Tools* 17(5):925–944.

Loutfi, A.; Coradeschi, S.; and Saffiotti, A. 2005. Maintaining coherent perceptual information using anchoring. In *2005 International Joint Conference on Artificial Intelligence, IJCAI-05*, 1477–1482.

Oh, S.; Russell, S.; and Sastry, S. 2009. Markov chain monte carlo data association for multi-target tracking. *IEEE Trans. Autom. Control.* 54(3):481–497.

Persson, A.; Dos Martires, P. Z.; De Raedt, L.; and Loutfi, A. 2020. Semantic relational object tracking. *IEEE Trans. Cogn. Dev. Syst.* 12(1):84–97.

Reid, D. 1979. An algorithm for tracking multiple targets. *IEEE Trans. Autom. Control* 24(6):843–854.

Ristani, E.; Solera, F.; Zou, R.; Cucchiara, R.; and Tomasi, C. 2016. Performance measures and a data set for multi-target, multi-camera tracking. In *Computer Vision - ECCV 2016 Workshops*, 17–35.

Ruiz-Sarmiento, J. R.; Günther, M.; Galindo, C.; González-Jiménez, J.; and Hertzberg, J. 2017. Online context-based object recognition for mobile robots. In *2017 IEEE International Conference on Autonomous Robot Systems and Competitions, ICARSC 2017*, 247–252.

Tremblay, J.; To, T.; Sundaralingam, B.; Xiang, Y.; Fox, D.; and Birchfield, S. 2018. Deep object pose estimation for semantic robotic grasping of household objects. In *2nd Annual Conference on Robot Learning, CoRL 2018*, 306–316.

Wong, L. L.; Kaelbling, L. P.; and Lozano-Pérez, T. 2015. Data association for semantic world modeling from partial views. *Int. J. Robotics Res.* 34(7):1064–1082.

Wong, L. L. 2017. Learning the state of the world: object-based world modeling for mobile manipulation robots. *AI Matters* 3(1):21–22.

Zuidberg Dos Martires, P.; Kumar, N.; Persson, A.; Loutfi, A.; and De Raedt, L. 2020. Symbolic learning and reasoning with noisy data for probabilistic anchoring. *Frontiers Robotics AI* 7:100.