Visual Scene Detection and Interpretation using Encyclopedic Knowledge and Formal Description Logic

Dejan Pangercic, Rok Tavcar, Moritz Tenorth, Michael Beetz
Intelligent Autonomous Systems, Technische Universität München
{dejan.pangercic, tavcar, tenorth, beetz}@in.tum.de

***Abstract*— In this system paper we report on our experience while working with a top-down guided 3D CAD model-based vision algorithm, being executed by an autonomous robot on objects (tableware and cutlery) in an Assistive Household environment. Top-down guidance is shaped upon how-to instructions which are parsed and extracted from the *wikihow.com* webpage - one of the world's largest resource of natural language task descriptions. Therein we selected a *How to set a table* entry and thus constructed this paper upon conversely interpreting the table setting for a meal. The robot's knowledge base is represented in Description Logics (DL) using the Web Ontology Language, and the inferences are obtained by virtue of SWI-Prolog queries. The whole proposed system is controlled by a modern, leading-edge Reactive Plan Language (RPL) which is the basic planning feature in the Assistive Household.**

I. Introduction

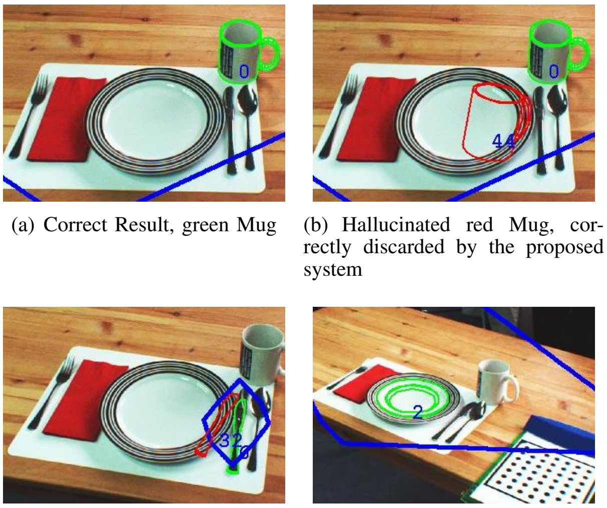
WHY Consider a household robot jointly setting the table with a human. To be helpful, the robot must be capable of looking at the table and inferring which items are missing, which items are not in the reach of the person who probably wants to use them, which items are misplaced, etc.

We believe that advanced scene perception mechanisms are essential for future personal robot applications. In this paper we investigate the realization of such a mechanism that is able to reason about and disambiguate percepts, i.e. images and video-streams. Our proposed system combines active vision with a formal knowledge base for advanced scene perception. The active vision component infers a task description, a region of interest (from the spatial configuration) and a set of relevant object categories based on instructions imported from the web (e.g. wikihow.com, see Listing 1).

```
Place the PLACEMAT in front of the chair.
Place the NAPKIN just left of the center of the placemat.
Place the PLATE(ceramic, paper or plastic, Ceramic preferred)
 in the center so that it just covers the right side of
 the napkin.
Place the FORK on the side of the napkin.
Place the KNIFE to the right so that the blade faces the
plate.
Place the SPOON right next to the knife.
Place the CUP to the top right corner of the placemat.
```

Listing 1. Instructions from http://www.wikihow.com/Set-a-Table, set of objects in capital, spatial relations in bold

The perceived camera image is then interpreted to detect the objects of interest and to localize them. The objects and their obtained locations are asserted to a factual knowledge base. Logical assertions in the form of rules (obtained from the transformed instructions from the web) then check properties of the scene such as "is **fork** on the **side** of the **napkin**".



(a) Correct Result, green Mug

(b) Hallucinated red Mug, correctly discarded by the proposed system

(c) Hallucinated red Knife, falsely accepted by the proposed system

(d) Falsely detected green Plate, discarded by the proposed system

Fig. 1. Possible situations upon search returns

For object detection, we use a highly prolific state-of-the-art 3D CAD model-based vision algorithm [1] which demands certain properties of the environment to be controlled and known (like the object's distance from the camera, possible orientations), and therefore *cannot* be successfully used without knowledge about the spatial configuration of the scene.

Fig. 1 denotes the possible situations, followed by a performed search:

The vision algorithm only detected the object correctly in case a). In case b), the red colored mug was hallucinated, but correctly discarded by the proposed system, knowing it can not stand in the air or be tilted this way. However, in c) the red colored knife was hallucinated and accepted due to the edge-like texture of the plate's boundary yielding a physically possible pose. Finally, in d) the plate was detected 0.2m below the table surface, which is a physically impossible pose.

These cases shall serve as motivation to perform the top-down guided visual search, in our case using a combination of common-sense knowledge and naive physics reasoning. In the remainder of the paper we demonstrate that by influencing the vision algorithm with only this information, while leaving other algorithm's parameters intact, the search results indeed improve substantially.

HOW After correctly parsing the natural-language web instructions (Listing 1), the proposed system converts them

[0] The research reported in this paper is partly funded by the German cluster of excellence CoTeSys (Cognition for Technical Systems).

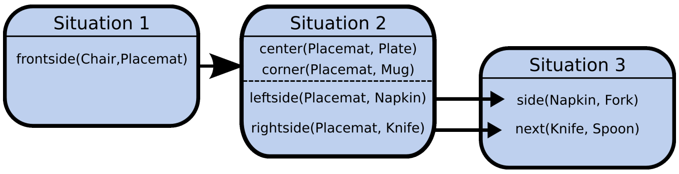


Fig. 2. Situation Graph Tree for Table Setting Scene

into a *Situational Graph Tree* [2] (SGT, see Fig. 2), which describes 3D objects of interest and their spatial relations as predicates in first-order logics.

These predicates are grounded in the perception data and used for top-down guidance when detecting objects, as depicted in Fig. 3. The search is performed using a 3D CAD model-based vision algorithm.

The algorithm requires the extreme poses of a sought object in the table coordinate system as a parameter. We dub this the *Cartesian Search Space* (CSS = mean pose plus search extent) restriction. From that, we generate two types of restrictions by means of a Modified Unscented Transformation (MUT) [3]. One is setting the algorithm's spherical search space parameters (3D restriction) and the other extracting an appropriate region of interest (ROI) from the search image (2D restriction, denoted with a blue contour in every image in this paper).

The central control unit is written in the Reactive Plan Language (RPL, [4]) and processes the SGT, infers objects to be found on the table, triggers the search and reasons upon objects' spatial relations. RPL can, in the current implementation, instruct the vision algorithm to perform two kinds of searches: one using a semi-restricted CSS (in the following: table search space) and second using a closely-restricted search space (in the following: object search space).

All matches are evaluated against both the hypothesis that they shall rest on top of another object (represented by the *on_Physical* relation), and the spatial relations described in the SGT. All matches of the vision algorithm which do not lie on the table surface or do not conform to the spatial relations described in the SGT (e.g. that a fork is to be placed on the "side" of the napkin) are deemed erroneous by the proposed system.

Finally, we perform an evaluation whether the complete table setting task was executed correctly, i.e. whether all required objects are present and conform to the WWW instruction. Based on the result, one could reason about the number of persons taking part in the meal, fetch missing objects, and also learn spatial properties like a usual distance for the "on the side of the napkin" relation.

Our Contribution We propose a novel, first-time achieved approach by connecting a high-level RPL control system (which encapsulates common sense knowledge processing and representation) and low-level visual sensing. We are able to model an every day situation scene as a first-order DL problem by retrieving and converting natural-language task instructions from the WWW.

In a proof-of-concept manner we manifest our approach by detecting a realistic table setting scenario by making use of a 3D CAD model-based detection algorithm which we support such that a substantial amount of falsely detected objects gets filtered out. Our approach is by no means limited to the indicated scenario only, in the work of [5] it has already been shown that a multitude of other instructions (e.g. "How to make Toast") can be retrieved from the natural-language instructions as well. Given a wealth of existing perception algorithms, from which some can fetch object information from internet databases, e.g. [6], we are therefore confident that the detection of even more complex and spatially correlated scenes will become feasible through an extension of herein presented work.

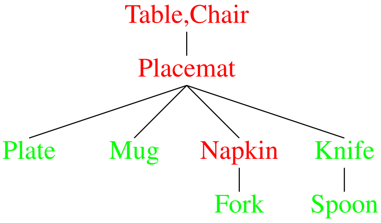


Fig. 3. Temporal hierarchy of sought objects (green), objects coded red are not part of the search

The remainder of the paper is structured as follows. In Sec. II, similar approaches and inspiration for our work are presented. The overall architecture of the proposed system is presented in Sec. III; a more detailed description of our methods can be found in Sec. IV and Sec. V. Results of the extensive tests are showcased in Sec. VI, discussed in Sec. VI-C and final remarks and future work are conveyed in Sec. VII.

II. Related work

We divide our literature overview into 2 parts, one regarding related systems making use of common sense knowledge models, and the other one regarding vision-driven object detection algorithms.

In recent years, a growing interest in artificial cognitive systems has brought about increased efforts to extend the capabilities of computer vision systems towards higher-level interpretations. They mostly consider optical flows in image sequences and represent the trajectories in another feature space (e.g. multivariate observation vector) [7], [8], [9], [10].

Arens and Ottlik [2] have been one of the first to demonstrate with concrete experiments in the street traffic domain that high-level hypotheses about intended vehicle behavior could in fact be used to influence the tracking unit and thus improve tracking under occlusion. They also present a concept of the so-called SGT which we partially adopt in our work. Neumann and Möller [11] present a concept of aggregates composed of multiple parts and constrained primarily by temporal and spatial relations. It is shown that these can be used to represent high-level concepts such as object configurations, occurrences, events and episodes. Their scene interpretation is modelled as a stepwise process which exploits taxonomical and compositional relations between aggregate concepts, while incorporating visual evidence and contextual information. The aggregates are represented in a $\mathcal{ALCF}(\mathcal{D})$ Description Logic related to the one presented in this paper. However, we point out that it amounts to manual modelling of the expected scene beforehand as opposed to ours that does it automatically. Further, Hotz et. al. [12]

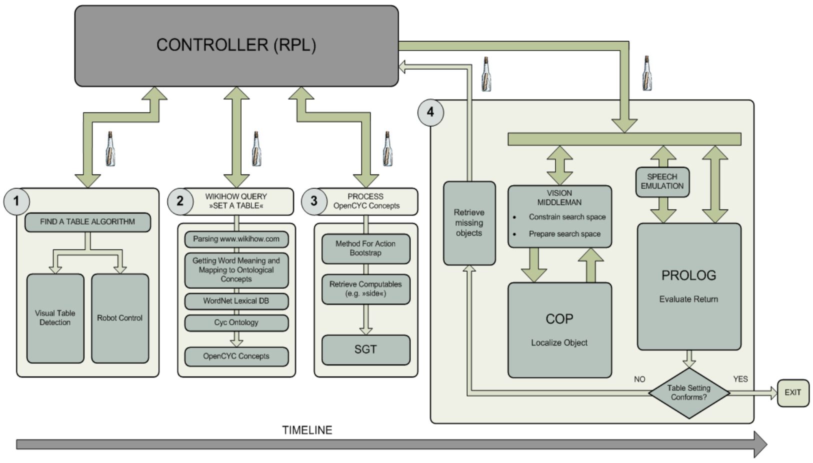


Fig. 4. Overall System Architecture

argue that the generic high-level conceptual units introduced by Neumann can generate feedback in a generic manner. Demonstrations are carried out using SCENIC, a system capable of part-whole reasoning, i.e. establishing an aggregate instantiation based on evidence for any of the aggregate parts. It allows to generate strong expectations about further evidence for parts of this aggregate and to feed back these expectations to lower levels. While existing approaches to high-level interpretation differ in many aspects, they share the commonality that prior knowledge about spatial and temporal relations between several objects has to be brought to bear. Baiget et al. [13] however were the first in trying to automatize the geometrical construction of a scene by learning it from tracking humans. Learning is done by means of FMTHL (Fuzzy Metric Temporal Horn Logic) which also generates conceptual predicates from the state vector.

In the overall picture, we aim at the *detect-approach-grasp* system which allows for smooth integration into the Assistive Household. Inspired by that when selecting the object detection algorithm, we set for 3 criteria: the algorithm is to detect objects in 3D, use texture and/or shape information, and is to feature generic characteristics (e.g. arbitrary type of object). There are several proposed methods [14], [15], [16] allowing for that; however, they all require several views of the object to determine its pose, as opposed to the 3D CAD model-based algorithm of our choice that infers poses from a single image [1].

III. System Architecture

The architecture on display in Fig. 4 presents a large and software-wise very complex integrated approach towards context-rich scene detection in human environments like an Assistive Household [17]. The proposed system consists of 3 main functional units, each performing on a different architectural level: A high-level RPL controller (grey), the YARP middleware (peripheral connections) [18], and end-processing modules (enumerated 1 through 4).

The latter perform the following tasks:

1) Finding the kitchen table,
2) Parsing and converting the natural-language task instructions ("How to set a table") into a parsable tree of logic assertions,
3) Transforming the logical assertions into the SGT, and
4) Object localization and interpretation of results.

Detailed descriptions of the modules 2), 3), 4) are provided in the following two sections and denote our major contributions.

IV. Transforming How-tos into SGTs

A. Translating Natural Language Instructions

This module, depicted in Block 2 in Fig. 4, transforms natural-language task instructions as the one in Listing 1 into a set of logical assertions. These assertions represent the sequence of actions with the respective objects, actions, pre- and postconditions, temporal constraints, quantifiers and prepositional relations.

Originally proposed by Tenorth et al. [5], the procedure will only be briefly recapped here. After the syntactical structure of the sentences is determined by a probabilistic context-free grammar (PCFG) parser, the words' senses are resolved using the WordNet lexical database and the Cyc ontology [19]. The content of our robot's knowledge base is derived from Cyc, representing objects in the environment as instances of the respective Cyc classes. Technically, the knowledge is stored as *Classes*, also referred to as concepts, and *Instances* in Description Logics using the Web Ontology Language (OWL).

The class level contains abstract terminological knowledge like the types of objects, events and actions (e.g. KitchenTable), organized in a taxonomic structure. Instances represent concrete physical objects (e.g. kitchentable1) or actually performed actions. Properties, also called Roles, link instances. All relations are formulated as *(Subject, Property, Object)* rdf_triples.

An example outcome of the transformation procedure, a conveniently parsable list of instructions, is available for display at `http://www9.cs.tum.edu/people/pangercic/files/web.txt`.

B. Generating the Situation Graph Tree

The compound logical assertions obtained in Sec. IV-A are next transformed into code appropriate for use within the RPL program (Fig. 4, Block 3). A detailed description of the knowledge processing system can be found in [20].

Given that our proposed system is to answer questions like "Which objects are missing and what are their spatial relations to the found ones?", we decided to convert the logical representation into the SGT as proposed in [2] (see Fig. 2).

For the sake of clarity, let us first introduce the terminology used in the formal instruction representation:

- *methodForAction* - root predicate listing all actions in a How-to
- *actionSequence* - predicate specifying a sequence of actions
- *objectActedOn* - link between an action and the main object it is performed on
- *purposeOf-Generic* - predicate representing postconditions of an action
- *PuttingSomethingSomewhere* - action denoting "taking something to"
- *[in, to, on, next]-Underspecified[*]* - rather general prepositional relations
- *parts-Underspecified* - predicate associated with the preposition "of"

The logic-based representation of an action sequence, like the tablesetting task, looks like the following:

```
(methodForAction
 (Set-a-Table kitchentable1)
  (actionSequence
     (TheList #$puttingsomethingsomewhere1
                     ...
                     ...
                     #$puttingsomethingsomewhereN)))
```

To build an SGT, we start with the *methodForAction* predicate that is the root of the formal task specification. In the case of "Set-a-Table", the action sequence mainly consists of instances of the *PuttingSomethingSomewhere* class. Each of these actions is required to pass a completeness test that check for instance if the object to be manipulated and the target location are given.

To exemplify, let us find a relation for an object of type *Fork-SilverwarePiece*: The *objectActedOn* relation specifies which object the action *put1* of type *PuttingSomethingSomewhere* is to be executed on. *PurposeOf-Generic* is used to describe post-conditions; in this case, the outcome of *put1* shall be that the object *fork-silverwarepiece1* is related to *Side* by the *on-UnderspecifiedSurface* relation. The English word "on" was mapped to the very general *on-UnderspecifiedSurface* relation during the translation step; now, the "on the side" expression has to be resolved properly.

Since *Side* is no physical object that the system can detect in the scene, the resolution process continues with the *parts-Underspecified* predicate. This very general relation is statically mapped to the English word "of" and, in this case, points to the object *Napkin*. The result is thus that an object of type *Fork-SilverwarePiece* is to be put on a *Side* of a *Napkin*. The result of this procedure is the SGT in Fig. 2.

For checking if the spatial relations described in the how-to hold in the situation at hand, we use *computable* predicates [20]. These predicates are computed on demand during the reasoning process and allow, one the one hand, to easily load external data into the knowledge base, and, on the other hand, to calculate spatial relations based on object locations.

The *side* property, which holds if the object is on one side of the subject, is one example of such a computable property. If the relation holds can be checked with the following query:

```
?- owl_query('side', fork1, ?A)
A = 'napkin1'
```

Internally, this query is translated to a call to a function that compares the observed object coordinates and, using a heuristic, determines if they are next to each other.

V. Top-Down Guided Object Search and Interpretation

When sorting out all required pre-processing, we commence the actual search for objects as denoted in Block 4 in Fig. 4.

The RPL controller extracts the set of objects and computable predicates from the SGT, converts predicates into CSSes and sends them to the Vision Middleman (VM). The VM processes the respective CSS and sends it to a larger framework called COP, which generates 2D and 3D search spaces to be used with the 3D vision algorithm.

In our case, we apply two CSSes: *table search space* and *object search space*. They are both intuitively limited and processed using MUT as indicated in Sec. I. The object search space limitations are determined based on the spatial configuration from the natural-language instruction. Search space extents are generated according to sizes of the sought-after objects, and mean poses are set heuristically. It is on our future research agenda to amend the latter by learning the spatial relations from observations of humans and thus preserve the generic character of this approach.

After COP has completed the search for objects, the estimated object poses are returned to RPL, which updates their poses in the knowledge base by calling the *entityLocation(Pose)* routine.

Finally, the correctness of the result of the table setting task is tested with the following first-order predicate calculus:

$$\begin{aligned} [frontside(Chair, ?Placemat) \rightarrow \\ on\_Physical(Table, ?Objects) \\ \wedge\, center(Placemat, ?Plate) \wedge corner(Placemat, ?Mug) \\ \wedge\, \{leftside(Placemat, Napkin) \rightarrow side(Napkin, ?Fork)\} \\ \wedge\, \{rightside(Placemat, ?Knife) \rightarrow next(Knife, Spoon)\}] \\ \rightarrow \textbf{isValidTableSetting} \end{aligned} \tag{1}$$

The table setting is therefore deemed to be correct if all operands (computable predicates) return a value *true*.

In this work, we employ only the 3D CAD based detection algorithm. The objects *Placemat* and *Napkin* cannot be detected using this algorithm, therefore their poses are hard encoded into the world model and the predicates "frontside" and "leftside" in the SGT thus always return *true*.

The outcome of the calculus from Eq. 1 can lead to the following results. All cases are visualized in Fig. 1.

- True Positive (TP): correct that the object corresponds to the how-to manual and on_Physical.
- True Negative (TN): correct that the object does not correspond to the how-to manual and on_Physical.
- False Positive (FP): incorrect that the object corresponds to the how-to manual and on_Physical.
- False Negative (FN): incorrect that the object does not correspond to the how-to manual and on_Physical.

VI. Empirical Evaluation

In order to evaluate our approach we apply our proposed system to a set of ten different realistic images[1] of a table setting in the Assistive Household. The images comprise 10 different views of the robot onto the table surface, where tableware and cutlery for 1 person meal are correctly placed. The aim of the test was to validate the correctness of the setting through an interactive two-steps procedure (once for *table search space* and once for *object search space*).

The table coordinate system (CS) was determined by means of an artificially inserted calibration plate on one corner of the table. The z axis of the coordinate system is pointing upwards, the x and y axes are aligned with the table's shorter and longer edge respectively. We are aware of the crudeness of this choice for the general case of the Assistive Household, however to prove the concept we needed to exclude all additional sources of errors that influence to the final localization result yielded by the vision algorithm.

A. Objects Detection Using Table Search Space, 1st run

Fig. 5 shows results of the query to find a maximum of five respective instances of objects from Fig. 3 in every of previously mentioned ten images. It is evident that the search is over-dimensioned and, due to the nature of the search algorithm, also incorrect in number of returned matches and their localization.

Table I, row 1 presents this quantitatively. In three images we correctly found only one object, while we never detected all five objects. In overall, 22 TP, 165 TN, 0 FP and 19 FN results were obtained (50 TPs were expected), which shows that a great deal of hallucinated objects were successfully rejected. However, due to the textually rich table surface, too few actual objects were found with high enough score.

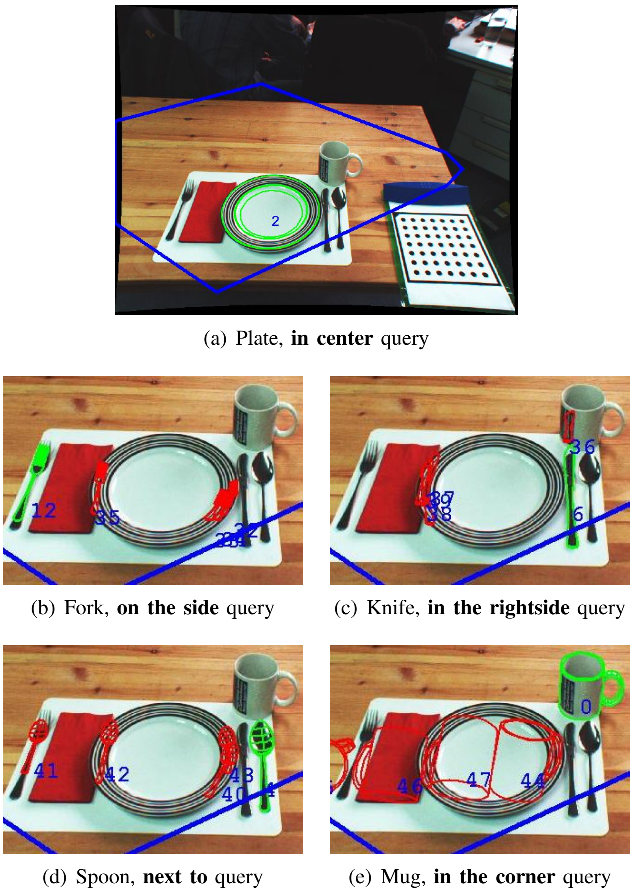
#ofObjectsFound	One	Two	Three	Four	Five
#ofImages - 1st run	3	3	3	1	0
#ofImages - 2nd run	0	1	1	5	3

TABLE I

Nr. of objects found in nr. of Images

B. Objects Detection Using Common Sense Knowledge Object Search Space, 2nd run

In the second run, we perform another search with the search space limited to near vicinities of the expected poses of sought-after objects (object search space). Fig. 6 denotes this situation. By not allowing the algorithm to take viewpoints which, due to background texture, would yield hallucinated matches with better scores than the real objects, we obtain more TPs and better localization rates. On the other hand, however, we obtain some matches in the direct vicinity of the true match which consequently leads to FPs. In overall, the search yielded 40 TPs, 42 TNs, 4 FPs and 10 FNs. Table I, row 2 shows a substantial improvement in terms of overall performance. In five images we were able to successfully detect four out of five objects, and in three images even the full set (all five objects) was detected.



(a) Plate, **in center** query

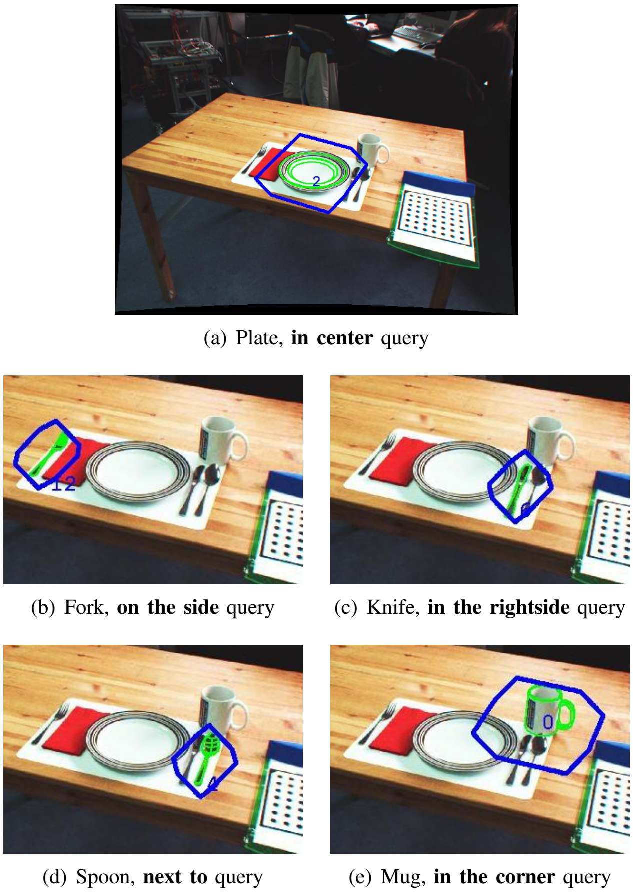
(b) Fork, **on the side** query

(c) Knife, **in the rightside** query

(d) Spoon, **next to** query

(e) Mug, **in the corner** query

Fig. 5. Best results of querying for objects in table surface search space. The search space is overdimensioned and yields ample of erroneous matches.



(a) Plate, **in center** query

(b) Fork, **on the side** query

(c) Knife, **in the rightside** query

(d) Spoon, **next to** query

(e) Mug, **in the corner** query

Fig. 6. Best result of querying for objects in search space adjacent to the expected poses of objects. The amount of erroneous matches (coded red in Fig. 5) dropped substantially.

[1] The data set is available for public use: http://www9.in.tum.de/people/pangercic/images/scene.tar

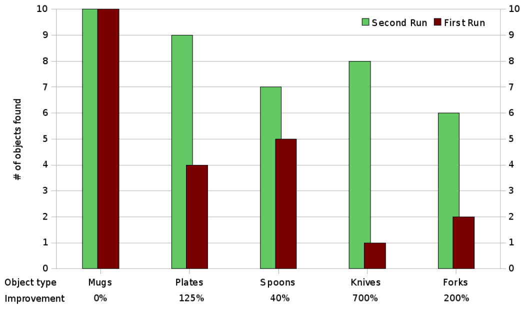


Fig. 7. Improvement of Object Detection by Further Restriction of Search Space, red - number of objects found in table search space, green - number of objects found in object search space

C. Recapitulation on Object Detection

By automating the search space restriction process we are able to use the 3D CAD based algorithm on-the-fly. We only set a subset of its parameters beforehand once, whereas the essentials (the spherical search space restriction) are automatically adapted upon changes in the image scene.

Since the used 3D CAD algorithm is oriented towards industrial applications, where conditions of the environment can be controlled, we had to tackle numerous challenges while adapting its use to our situation. While tableware is relatively unproblematic, detecting silverware is a challenge partly because of its smaller size. At usual distances of $[1m, 2m]$, tableware is salient in the image, while silverware's shape is captured poorly. Therefore, we must exclude sources of hallucinated matches (by setting the ROI) on one end and reason upon validity of detection results (using the on_Physical relation) on the other. The table search space is still too large to solve that problem. Using common sense knowledge obtained from the WWW, we are able to restrict the search space even further and thus boost the detection algorithm's success rate, especially for silverware objects. Detection of the latter proves to otherwise be of inadmissibly low quality. Another problem regarding silverware, which our proposed system does not tackle, is its reflective surface which yields erroneous results under sub-optimal lighting conditions. For that, our future agenda includes using CCD cameras in combination with other sensors. The chart in Fig. 7 shows the improvement of the object detection rates after a further restriction of the Cartesian search space. For all objects but Mugs, which were always detected in both cases, detection improved substaintally, in overall by 82%. The calculation times improved as well.

VII. Future Look

The immediate task on our research agenda is related to learning of spatial relations. It is envisioned to filter out false matches only by the on_Physical relation and cluster the remaining ones. In an iterative loop, the mean poses of these clusters will be validated against the predicates from Fig. 2, and the matches will be re-clustered. By that we will free the proposed system of stark hypotheses for objects' expected poses. Further, as already indicated in the work of [21], data fusion between a CCD camera, a time-of-flight camera and a laser scanner will be exploited in order to overcome peculiar characteristics of particular objects (e.g. shiny silverware).

References

[1] C. Wiedemann, M. Ulrich, and C. Steger, "Recognition and tracking of 3d objects," in *Pattern Recognition*, ser. Lecture Notes in Computer Science, G. Rigoll, Ed., vol. 5096. Berlin: Springer-Verlag, 2008, pp. 132–141.

[2] M. Arens and A. Ottlik, "Using behavioral knowledge for situated prediction of movements," in *In: Proc. 27th German Conference on Artificial Intelligence (KI-2004). Volume LNAI 3238*. Springer, 2004, pp. 141–155.

[3] R. Tavcar, "Connecting high-level planning, reasoning and model-driven vision into a robotic system that enables everyday manipulation tasks," Master's thesis, Technische Universitaet Muenchen, 2009.

[4] D.McDermott, "A reactive plan language," Yale University," Research Report YALEU/DCS/RR-864, 1991.

[5] M. Tenorth, D. Nyga, and M. Beetz, "Understanding and executing instructions for everyday manipulation tasks from the world wide web," IAS group, Technische Universität München, Fakultät für Informatik, Tech. Rep., 2009.

[6] U. Klank, M. Z. Zia, and M. Beetz, "3D Model Selection from an Internet Database for Robotic Vision," in *International Conference on Robotics and Automation (ICRA)*, 2009.

[7] D. Thirde, M. Borg, J. Ferryman, F. Fusier, V. Valentin, F. Bremond, and M. Thonnat, "A real-time scene understanding system for airport apron monitoring," in *ICVS '06: Proceedings of the Fourth IEEE International Conference on Computer Vision Systems*. Washington, DC, USA: IEEE Computer Society, 2006, p. 26.

[8] S. Gong and H. Buxton, "Understanding visual behaviour, special issue introduction," vol. 20, no. 12, pp. 825–826, October 2002.

[9] R. Howarth and H. Buxton, "Conceptual descriptions from monitoring and watching image sequences," vol. 18, no. 2, pp. 105–135, January 2000.

[10] M. Brand, L. Birnbaum, and P. R. Cooper, "Sensible scenes: Visual understanding of complex structures through causal analysis," in *AAAI*, 1993.

[11] B. Neumann and R. Möller, "On scene interpretation with description logics," in *Cognitive Vision Systems: Samping the Spectrum of Approaches*, ser. LNCS, H. Christensen and H.-H. Nagel, Eds. Springer, 2006, no. 3948, pp. 247–278.

[12] L. Hotz, B. Neumann, and K. Terzic, "High-level expectations for low-level image processing," in *KI '08: Proceedings of the 31st annual German conference on Advances in Artificial Intelligence*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 87–94.

[13] P. Baiget, C. Fernández, X. Roca, and J. Gonzàlez, "Automatic learning of conceptual knowledge in image sequences for human behavior interpretation," in *IbPRIA '07: Proceedings of the 3rd Iberian conference on Pattern Recognition and Image Analysis, Part I*, 2007, pp. 507–514.

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

[15] G. Kootstra, J. Ypma, and B. de Boer, "Active exploration and keypoint clustering for object recognition," *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pp. 1005–1010, May 2008.

[16] I. Gordon and D. G. Lowe, *What and Where: 3D Object Recognition with Accurate Pose*, ser. Lecture notes in Computer Science. Springer Berlin/Heidelberg, 2006, vol. 4170.

[17] M. Beetz, F. Stulp, B. Radig, J. Bandouch, N. Blodow, M. Dolha, A. Fedrizzi, D. Jain, U. Klank, I. Kresse, A. Maldonado, Z. Marton, L. Mösenlechner, F. Ruiz, R. B. Rusu, and M. Tenorth, "The assistive kitchen — a demonstration scenario for cognitive technical systems," in *IEEE 17th International Symposium on Robot and Human Interactive Communication (RO-MAN), Muenchen, Germany*, 2008, invited paper.

[18] G. Metta, P. Fitzpatrick, and L. Natale, "Yarp: Yet another robot platform," *International Journal of Advanced Robotics Systems, special issue on Software Development and Integration in Robotics*, vol. 3, no. 1, 2006.

[19] "OpenCyc," 2009, `www.opencyc.org` .

[20] M. Tenorth and M. Beetz, "Towards practical and grounded knowledge representation systems for autonomous household robots," in *Proceedings of the 1st International Workshop on Cognition for Technical Systems, München, Germany, 6-8 October*, 2008.

[21] Z. C. Marton, R. B. Rusu, D. Jain, U. Klank, and M. Beetz, "Probabilistic Categorization of Kitchen Objects in Table Settings with a Composite Sensor," in *Submitted to the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2009.