Web-enabled Robots
Robots that use the Web as an Information Resource

Moritz Tenorth, Ulrich Klank, Dejan Pangercic, and Michael Beetz
Intelligent Autonomous Systems, Technische Universität München
{tenorth, klank, pangercic, beetz}@cs.tum.edu

***Abstract*— As robots are starting to perform everyday manipulation tasks, such as cleaning up, setting a table or preparing simple meals, their control programs must become much more knowledgeable than they are today. Typically, everyday manipulation tasks are specified vaguely and the robot must therefore infer by itself how to do the appropriate actions to the appropriate objects in the appropriate way in order to accomplish these tasks. These inferences can only be done if the robot has access to the necessary knowledge, including knowledge about how objects look, what their properties are, where they can be found, what might happen if particular actions are performed on them, etc.**

In this article, we describe and discuss the use of information that is available in the world-wide web and intended for human use as a knowledge resource for autonomous service robots. To this end, we introduce several categories of websites that can serve as information sources and explain which kinds of information they provide. We then investigate several information processing methods that can access these websites in order to provide robots with necessary knowledge for performing everyday manipulation tasks. The use of the web as a knowledge resource is a promising alternative to the hard and tedious task of coding comprehensive specific knowledge bases for robots.

I. Introduction

Performing complex everyday manipulation tasks, such as setting the table, cleaning up, and preparing meals, requires robots to have *plans* — robot control programs that can not only be executed but also explicitly reasoned about and manipulated [18], [3]. A large research community in AI planning is investigating the automatic generation of plans by studying the computational problem of computing action sequences that transform states satisfying a given state description into another state that satisfies a given goal [19].

We agree with McDermott [18] when he argues that this computational problem turned out to be too hard and too easy at the same time. The hardness of the problem is caused by its generality making the problem computationally unsolvable or intractable at best. At the same time, the problem oversimplifies its real counterpart, because action sequences are not expressive enough to specify competent robot manipulation behavior.

The world-wide web provides us with promising alternatives to reconsider the problem of realizing competent plan-based robots more successfully. Instead of generating plans in the classical way, *web-enabled robots* can make use of websites like *wikihow.com* to look up the appropriate courses of action on the web. After having read the instructions and transformed them into a plan, the robot must find the described objects and tools to perform the task. To do so, it can use other websites such as online shops to find out how for instance a bottle of pancake mix looks. The robot also has to know the properties of objects, for example if they are perishable, in order to decide where to search for them or how to handle them. Such information can also be found on the web, often on the same page as the product picture.

Fig. 1. Performing complex tasks like making pancakes requires a lot of knowledge. We present some approaches to acquire different kinds of knowledge from public web resources and make them available to robots.

The web-enabled generation of robot activity plans is attractive because web instructions already include sequences of actions and thereby simplify the plan generation problem. In addition, the instructions contain information about how actions should be carried out, hints about how to improve action execution, and potential problems. These pieces of information are necessary for producing more competent manipulation behavior.

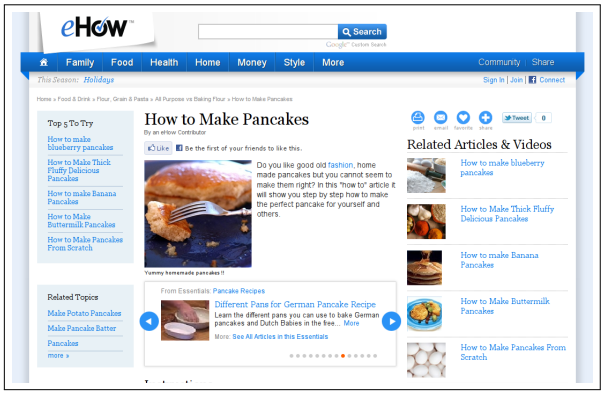
The web thus provides plenty of knowledge a robot can use to accomplish everyday tasks (Figure 2). *ehow.com* and *wikihow.com* contain thousands of step-by-step instructions for everyday activities like cracking and separating an egg, cooking different types of omelets, etc: about 92,000 on *wikihow.com* and more than 1.5 million articles on *ehow.com*. Lexical databases like *wordnet.princeton.edu* group verbs, adverbs and nouns semantically into sets of synonyms (synsets), which are linked to concepts in encyclopedic knowledge bases like *opencyc.org*. These encyclopedic knowledge bases, which are represented in a variant of first-order logic, contain an abundance of knowledge about concepts such as eggs. They can inform the robot about the nutrition facts of eggs or tell it that eggs are products of birds and fish. However, they typically lack action-related informa-

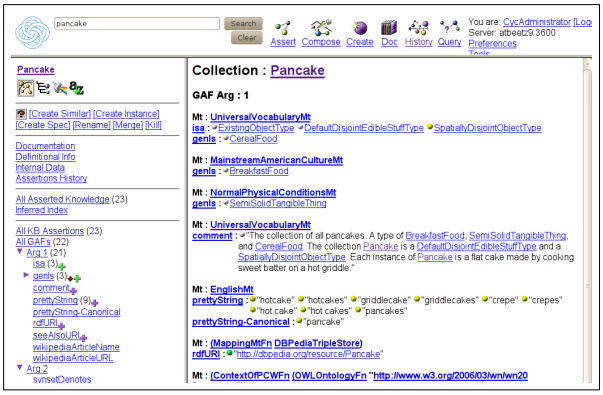
Actions in a task

- `ehow.com`, `wikihow.com`
- Step-by-step instructions for everyday tasks

Ontological relations

- `opencyc.org`
- Very large encyclopedic knowledge base



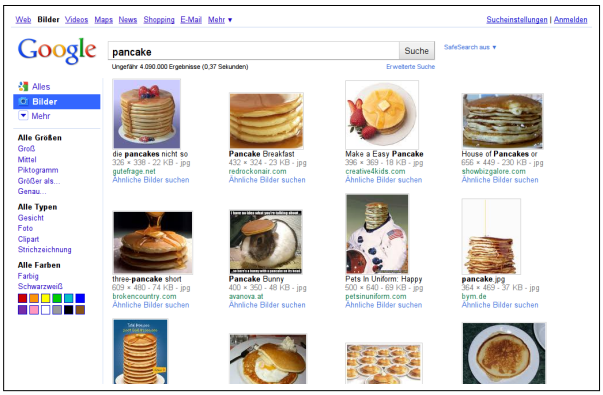


Common-sense knowledge

- `openmind.hri-us.com`
- Common-sense knowledge from Internet users

Object appearance

- `germandeli.com`, `images.google.com`
- Pictures of products and other object classes

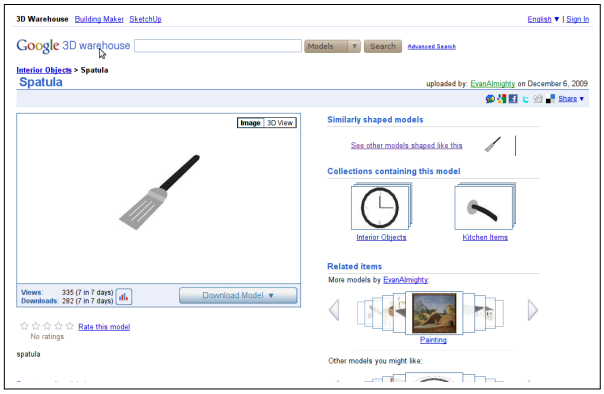


Object shape

- `sketchup.google.com/3dwarehouse/`
- 3D CAD models of household items

Object properties

- `germandeli.com`
- Object properties extracted from shopping websites



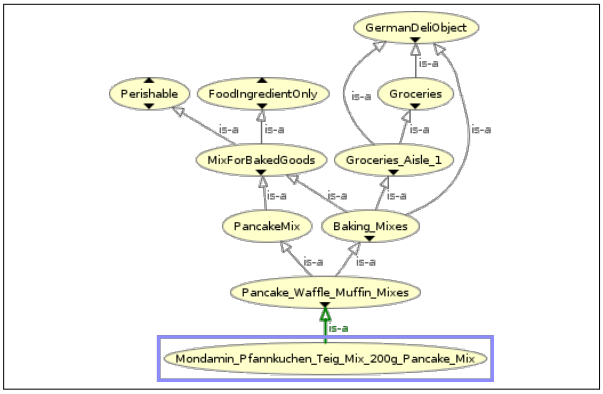


Fig. 2. Examples of web pages that provide useful information for a household robot.

tion, e.g. the piece of information that eggs can break easily and that they should be cooked or baked before consumption. This kind of knowledge is available at other websites such as the OpenMind Indoor Commonsense website (*openmind.hri-us.com*). But the web not only contains knowledge about object usage. A robot could also retrieve appearances of objects (*images.google.com*, *germandeli.com*) and even geometric models (*sketchup.google.com/3dwarehouse*).

There are two main contributions of this article: First, we give an overview of our approaches to enable robots to use web information to accomplish more tasks in a more general, flexible, and reliable manner, and to scale towards more realistic everyday tasks. In addition, we discuss different sources of knowledge on the web and our experience with using them: Which information do they provide, how can they be used for robot tasks, and which information is still hard to find on-line?

The remainder of this paper is organized as follows: We first describe a usage scenario (Section II) and discuss general problems that arise when trying to use web information, which was originally created for humans, in a robotics context (Section III). Then, we describe how different kinds of knowledge can be acquired, namely encyclopedic (Section IV) and common-sense knowledge (Section V), task instructions (Section VI), object recognition models (Section VII) and formal descriptions of object properties (Section VIII). We conclude with a discussion of what has been achieved in the area and where we think more research is required (Section IX).

II. Usage scenario

Over the course of the following sections, we will use the task of making pancakes as an example to explain the use of web information. A video of a recent live demonstration, in which our robots performed this task using knowledge that has been, to a large degree, acquired from web sources as described in the following sections, can be found in our YouTube channel[1]. Though extremely simple for a human, such tasks are very complex from a robotics point of view and require a lot of knowledge to be accomplished. The first part of the video thus shows, in form of a dialog with a human, which knowledge the robot needs for the task and where it can get it from.

In the beginning, the robot is asked to make pancakes. It looks up instructions on the web, finds some at *wikihow.com* (see Figure 3), and starts to translate them into an executable plan. In the first step, it parses the instructions, written in natural language, to identify the parts of speech and the sentence structure. Then it builds an internal representation of the actions described, resolves ambiguities in the description using its encyclopedic (Section IV) and common-sense knowledge (Section V), and generates a plan (Section VI).

Next, the robot checks which objects are required for the task – which can easily be done using the formal task description generated in the previous step – and whether it has

[1]http://www.youtube.com/watch?v=4usoE981e7I

Fig. 3. Example of instructions for making pancakes from wikihow.com

a recognition model for each of them. The example object, a bottle of instant pancake mix, was found on *germandeli.com*, a shopping website, and a recognition model was created from the product picture (Section VII and Figure 4).

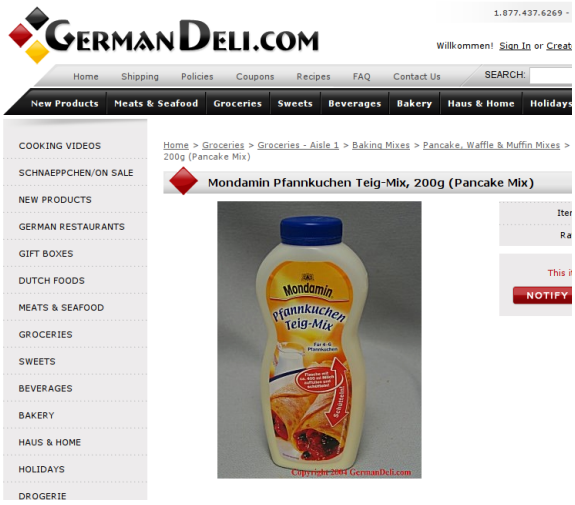


Fig. 4. Picture of a bottle of pancake mix obtained from an online shop.

Once the robot knows which objects it needs and how they look like, it has to find them in the kitchen environment. Since it has not only downloaded the picture of the pancake mix, but also information about its properties, it can infer that it needs to be stored in the refrigerator. Figure 5 illustrates the inference steps that are performed. The reasoning process combines encyclopedic knowledge about the fridge (upper left part), common-sense knowledge that a refrigerator is the storage place for perishable goods, spatial knowledge describing an instance of a refrigerator at a certain location in the environment, and knowledge about the pancake mix that was automatically generated from an online shop's website (Section VIII). To combine these different kinds of knowledge from different sources, a common representation is crucial. In our system, all knowledge is formally represented in the KnowRob knowledge base [35].

The methods presented in this paper helped the robot to

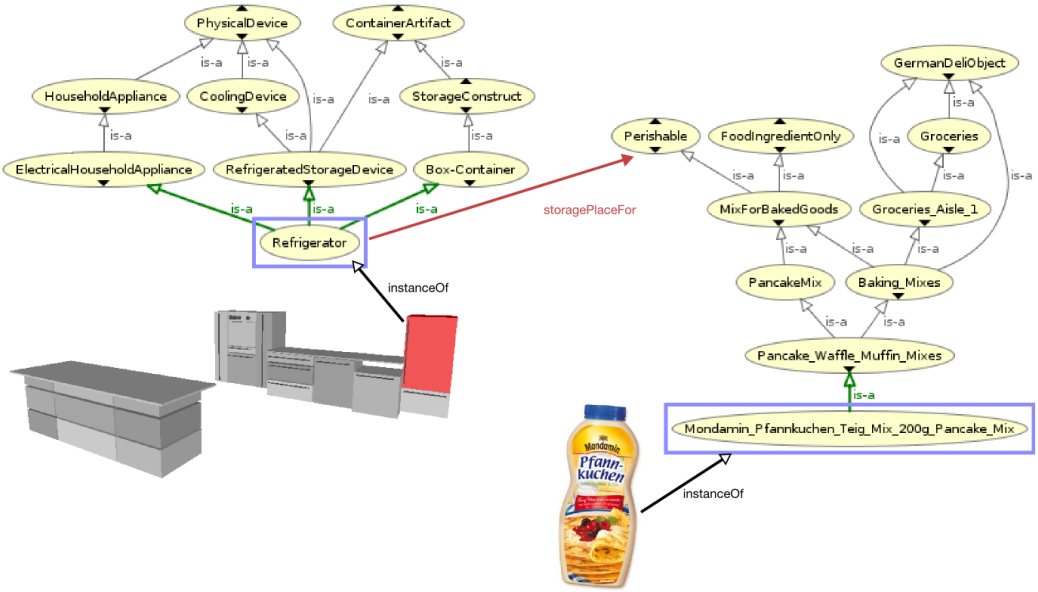


Fig. 5. Reasoning steps to infer a probable storage location for a bottle of pancake mix.

acquire important information. The methods for acquiring the knowledge worked well – the major problem is a lack of knowledge sources that provide the required information. We pushed the automatic import from existing sources as far as we could, but still had to manually add some scenario-specific routines or knowledge. For example, the robot's knowledge about actions, objects, and processes is still not deep enough to competently understand instructions written for humans. The instructions in the experiment refer to the "pancake mix", a liquid stuff-like substance (i.e. something that can be divided into smaller pieces without changing its type). While the robot can map this description to the correct concept in its knowledge base, it would also have to infer that this stuff is usually in a container, that it thus has to search for this container, open it, pour the mix onto the pancake maker, and further estimate how much of the mix is required. These relations are still manually specified, mainly since there is no source of information providing such knowledge. Equipping robots with sufficiently deep knowledge about objects and actions in the household scenario remains an open challenge.

The execution of the pancake making task was based on low-level routines that were vision-guided, but manually coded, like the routine for flipping the pancake, or routines for e.g. the visual calibration of the spatula after it has been picked up. Inferring that such routines are needed (in this case because of the millimeter-precision required for flipping the pancake) and parameterizing them correctly is another task for which robots need much more knowledge than they currently have.

III. Making use of web information

On the one hand, the World Wide Web is the biggest resource of knowledge that has ever been available to a robot, consisting of billions of pages that cover a huge range of topics for many different audiences, and which are all, in principle, machine-readable: digital text, pictures, and videos. On the other hand, most web pages are intended to be used by humans – that is, they are written in different natural languages and in a way that humans find them convenient to read. This makes it difficult for machines to use the information because they first need to understand the meaning of the words and sentences in natural language.

The semantic web initiative [11] was founded to overcome this problem by creating a world wide web for machines. In the semantic web, information is encoded in machine-readable form instead of natural-language text, i.e. in a way that computers can retrieve, understand, relate and process the information in the documents. Briefly, the semantics of a document are not hidden in the text, but explicitly described in a logic-based format computers can understand. In theory, this allows computers to autonomously answer queries by searching the web for information. In practice, however, only a very small fraction of the information on the web is available in the Semantic Web or as web services [24], especially hardly any information required by autonomous robots in household environments. Therefore, we needed to develop techniques to translate the information from human-readable form – instructions in natural language, pictures, and 3D models of objects – into representations the robot can use – formally described knowledge, task descriptions and object models that can be used for recognition.

Current work in web-mining and information retrieval does not formally represent the retrieved information [1], does only mine information from semi-structured sources [38], or focuses on finding information that is relevant for humans rather than understanding its content [16]. In contrast, robots need a much deeper understanding of the data, for instance instructions given in natural language, to make use of the information, and need to convert it into formal representations to relate the web information to other pieces of knowledge.

Despite the lack of content, the *techniques* developed as part of the semantic web initiative have proven extremely

useful for robots to acquire, represent, and use semantic information. Much research has been done on topics such as understanding web sites, creating ontologies from web data, and especially on developing standardized languages (e.g. RDF [2] and OWL [20]), query and exchange formats (e.g. SPARQL [28] and OWL-S [17]), and reasoning engines (e.g. HermiT [21] and Pellet [32]). In our research, we use these semantic web tools to represent and reason about the robot's knowledge. Whenever possible, we also try to use knowledge that is already available in semantic web formats.

IV. Encyclopedic knowledge

Robots need to use information from many sources: The vision system recognizes objects, a mapping system builds an environment map, a human gives commands, and the robot loads knowledge from different web sites. For making use of this knowledge, a robot has to autonomously integrate these different pieces of information, i.e. it needs to represent them in a common format, a common *language*, that formally describes the information including its semantics.

A natural solution to such problems is to use an *ontology* that formally describes and defines things and their relations – descriptions like the ones that can be found in an encyclopedia. This kind of knowledge about classes of things, their sub-classes and properties is thus called *encyclopedic knowledge*. Examples of such knowledge are that a refrigerator is a container (i.e. it can contain other objects) and a sub-class of cooling device and electrical household appliance (i.e. can be found in a household and needs electricity to operate). Such encyclopedic knowledge defines the terms a robot can use to describe its world in, to put things into relation, and to perform reasoning. Having a large encyclopedic knowledge base is thus crucial for being able to autonomously acquire and interpret knowledge.

Already some decades ago, large projects like Cyc [13] or SUMO [22] were started to collect encyclopedic knowledge on a large scale and to build a general upper ontology. To this end, researchers manually encoded very large amounts of knowledge in a machine-understandable format, a variant of first-order logic. These knowledge bases have become huge, covering a wide range of phenomena. However, this increase in size comes at a cost: Inference becomes rather slow, and ambiguities are created by knowledge a robot hardly ever needs. For a robot, "center" will mostly be a spatial concept, not a position in American Football. Recent efforts tried to automate the construction of knowledge bases by extracting encyclopedic knowledge from sources like Wikipedia ([38], [34]), mainly focusing on structured pieces of information such as categories and info-boxes. However, they mainly contain information about people and historic events and are thus not directly useful for a robot.

Neither Cyc nor SUMO are specialized for robotics, but were developed with the intention of understanding texts. For robot applications, it is often desirable to have less broad but deeper knowledge of the domain the robot is working in, like descriptions of different grasps or the concept of a "manipulation position" as the location where the robot should stand to manipulate objects. This was the reason to develop specialized knowledge bases for autonomous robots. Examples of such robot knowledge bases are KnowRob [35] and ORO [12]. Both use semantic web technology to represent information, which facilitates the integration of different sources of knowledge.

V. Common-sense knowledge

Encyclopedic knowledge provides the robot with definitions of object classes and their properties but often lack action-related information: What to do with the objects, how to handle them? In the pancake example, the robot has to know that one should *use a spatula* to flip the pancake, *use a pancake maker* to make pancakes or watch out for problems like spilled liquids or broken eggs. All this *common-sense knowledge* is completely obvious to humans and therefore usually not explicitly described. Humans assume that their communication partner also has this kind of knowledge and therefore usually omit such "obvious" information when explaining something. Therefore, robots also need common-sense knowledge to understand instructions given to humans – either in direct dialog or in indirectly via web pages.

The problem is that, since it is obvious to humans, most of this knowledge is typically not written down: humans usually acquire it already in their early childhood. Therefore, such knowledge has to be collected specifically for robots. Instead of letting a small group of experts create a knowledge, projects like the OpenMind Common Sense [30] initiative collect such data from Internet users by presenting them incomplete sentences and letting them fill in the gaps. While the OpenMind project collects general common-sense knowledge, the OpenMind Indoor Common Sense project (OMICS [6]) focuses on the kind of knowledge required by robots acting in indoor environments.

The users' responses are saved in semi-structured form in a relational database as sentence fragments in natural language. Several projects have started to convert the information into representations that support reasoning, for instance ConceptNet [14] or LifeNet [31]. Kunze et al. [10] translated the knowledge from the sentences in natural language into a formal logical representation in the KnowRob knowledge base.

The problem of acquiring large amounts of common-sense knowledge is still an unsolved issue. "Crowdsourcing" the collection by distributing the task to voluntary Internet users helps to scale the system but creates other challenges: Ambiguities in natural language are hard to resolve and even harder when looking at the short sentence fragments provided by OMICS. Relations are also interpreted completely differently by different people: A sentence fragment like "if A, then B" is interpreted as either immediate and inevitable effect (switching on a dishwasher changes its state from "off" to "on"), long-term effect (switching on a dishwasher results in clean dishes) or as indirectly related consequence (loading a dishwasher causes dishes to be clean – if, what is omitted, the soap is filled in, the hatch is closed and the device is turned on), or even as "implies" (dishes are clean

if the dishwasher has been turned on). Furthermore, there are gaps in the provided knowledge due to the way it was collected: being presented a sentence with placeholders to be filled in, people tend to enter the most obvious information. Presenting the same template to many people thus does not guarantee better coverage; instead, obvious statements occur several times, less obvious ones hardly ever. Nevertheless, such common-sense databases are a very useful source of knowledge that can hardly be found elsewhere, and the translation into semantic networks or description logics turns them into a useful resource for autonomous robots.

VI. Task instructions

Instructions taken from web sites need to be translated from natural language into formal, logic-based representations, and finally into executable robot plans. A detailed description of the system we developed can be found in [36]; here, we will just outline the main steps of the translation process. In the beginning, the sentences are parsed using a common natural-language parser [9] to generate a syntax tree (Figure 6 left). The branches of the tree are then recursively combined into more and more complex descriptions to create an internal representation of the instruction (Figure 6 right) describing the action, the objects involved, locations, time constraints, the amount of ingredients to be used etc.

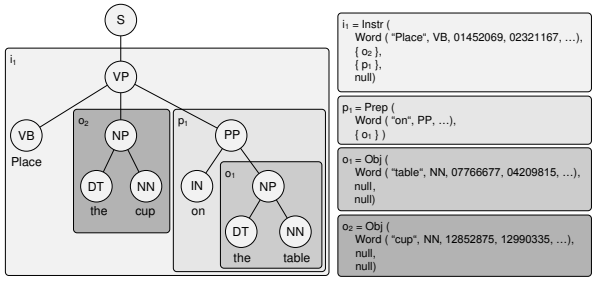


Fig. 6. Parse tree of an instruction (left) and resulting internal action representation (right). Part of our procedure for translating web instructions into executable robot plans [36].

The words in the original text are resolved to concepts in the robot's knowledge base by first looking up their meanings in the WordNet lexical database [5], and by exploiting mappings between WordNet and the Cyc [13] ontology. Usually, a word can have different meanings, so a disambiguation procedure needs to decide which one to choose. Currently, we use a rather simple method that is based on the phrase context and on information about object-action pairs obtained from Cyc. We are investigating how to improve this method by taking more of the robot's knowledge into account.

The translation system had originally been developed using instructions for setting a table or making tea (see [36] for the list of tasks and the conversion success rates). To use it in the pancake experiment, we only had to add a few mappings from WordNet to Cyc (e.g. for the pancake mix), the rest of the conversion process worked without modifications. Infering information that is missing in the instructions remains an open challenge: For instance, an instruction for setting a table states that items have to be put in front of the chair, but does not require them to be on top of the table. Other instructions fail to mention that an oven has to be turned off after use. Robots will have to detect these gaps and fill them appropriately – again, a very knowledge-intensive task.

Fig. 7. A sample image found on Google for `Barilla` (left) and an image taken with a consumer camera as a test case on the right side. Extracted features are visualized with green markers.

VII. Object recognition models

Having generated a plan for making pancakes, the robot has to find the right objects for the task. If the web instructions refer to objects the robot does not know about, it needs to acquire information about their appearance as well as their semantic properties. For packaged products, like the bottle of pancake mix, the recognition can use image features extracted from product pictures that can be found on shopping websites. For tools, e.g. the spatula for flipping the pancakes, the robot needs information about their shape that can for instance be found in databases of 3D CAD models.

A. Image search engines

A Google Images-based classification system can help recognize branded products. For a set of object classes, a training set of images can be obtained using a set of search terms, and a classifier that is able to distinguish those classes in images acquired by the robot can be trained in relatively low time. Google's mechanisms for finding pictures that are related to the most relevant pictures allow to acquire a reasonably good selection of training images.

The classifier is based on the "Bag of Visual Words" approach presented in [33]: Our implementation uses SIFT features extracted from a subset of the training images, and clusters the resulting SIFT vectors to get a discrete representation. This representation is referred to as the *codebook*, or the *Visual Words*, see Figure 7 for a sample image with extracted features. For performance reasons, the clustering is not directly performed on the whole training set but separately for all classes. The resulting clusters are then fused based on the Fisher criterion for two centroids of two clusters c_1, c_2 and their respective intra-cluster variance σ_1, σ_2:

$$d(c_1, c_2) > \frac{\|c_1 - c_2\|_2}{\sigma_1 + \sigma_2}$$

This enables us to apply this codebook to any search image after extracting the SIFT features in this image. We train a support vector machine for all relevant classes, which is then applied on the images acquired by the robot. False positive detections are reduced using the scores given by the support vector machine and thresholds on the minimum amount of features available in the images. To add spatial information,

Fig. 8. Samples from test on the left and training data on the right for the classes `Volvic` and `Coca Cola`.

we use geometric segmentation and restrict the extraction of features to object candidates in order to reduce clutter.

Tests showed that this method is able to achieve up to 95% accuracy for a reasonable number of classes (e.g., in one test, `Coca Cola, Fanta, Pringles, Milka, Barilla, Volvic`, see Figure 8 for a subset of the test and the training images for the classes `Volvic` and `Coca Cola`). This result was achieved with training data from the web and test data consisting of images of objects belonging to those classes taken under challenging illumination conditions.

B. CAD model databases

In order to manipulate objects, it does not suffice to know their types, a robot also needs spatial information. CAD models are one of the most accurate descriptions for rigid objects. Once a CAD model is available, a robot can recognize and localize the corresponding object in the environment and plan its manipulation actions.

The problem is that a robot usually does not have models for all objects it needs to manipulate. Assuming the class of the object is known, like *pan* or *cup*, we propose to download a set of CAD models from the Internet and to find the model that best fits the current sensor data (see [8] for details). The method requires that the system has a rough estimate of the location of the object (e.g. on top of the stove or inside a specific cupboard) and a good approximation of its size, e.g. from a segmentation of 3D sensor data or from common sense knowledge.

To make use of CAD models from the Internet for recognizing objects, a robot first has to select the most relevant objects to be downloaded and then needs to verify and compare the appearance of the actual object with the CAD model. The problem of finding relevant models is tackled using tags, which are assigned to the models by search engines like the Google 3D Warehouse. As part of the search results for words like *pan* or *cup*, we get a list of tags which allow to put the results into the right context. Usually, those tags describe the content of the CAD model better than its title or the fact that it was returned as an answer to a query. Models are selected from the result set if their tags have a low *semantic distance* to the search term [25]. This semantic distance is computed using the WordNet lexical database and describes how close the words are in the WordNet taxonomy. All selected models can be matched to the current scene and get a certain score based on the current sensor data and the respective method. If there is more than one good match, the system can interpolate between two 3D models to get an intermediate model which may fit the data even better than

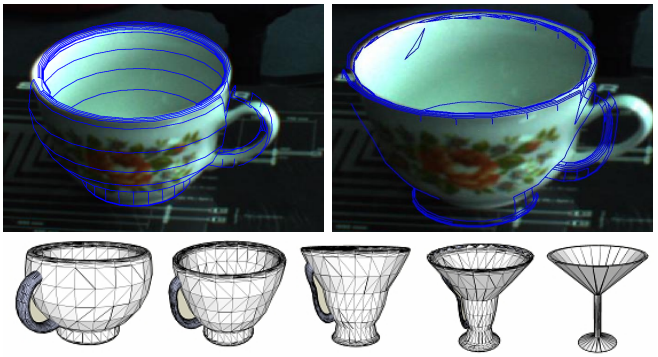


Fig. 9. On the lower image two models from the search for `cup` and a morphing between the two cups are displayed. The original model and an interpolated model are applied to the same scene to show that the interpolated model fits better in this case.

the original ones. This morphing procedure is explained in more detail in [39], an example is shown in Figure 9.

Once a model is selected, the system has to decide if this CAD model fits the sensor data from the initial position estimate. We integrated two methods into our system that are capable of taking an arbitrary CAD model and comparing it with sensor data of the expected position of the real object. The first method is based on geometric edges of the CAD model that are projected into a 2D image. Possible object poses are pre-calculated, the model is projected into a virtual edge image and compared in an efficient way to the image of the scene. This search is called ”3D Shape-Based Matching” and uses as core the method proposed by Ulrich et al. [37]. This method is evaluated in our setup with some improvement in [7]. The second method relies on 3D sensors. Instead of pre-calculating edges in the image, it computes the normals and the relations between sub-samplings of pairs of normals for the model, and compares them with the current 3D data using an efficient voting scheme. This method is based on the work proposed by Drost et. al. [4]. A model for this method can be extracted from a view or a CAD model of a 3D area-sensor. Both methods can suffer from false positives while not requiring a segmentation beforehand.

C. Online shops

Another possibility to make use of the World Wide Web for the recognition of household objects are online shops such as *germandeli.com*. Shopping websites contain more or less standardized descriptions of thousands of everyday objects including high-quality photos, all products are sorted into categories, and most pages have a very similar structure. This makes these websites an excellent source of information about manufactured products. Compared to the system for using the Google Images search (Section VII-A), which learns models of entire categories of objects, the method described here learns models of specific objects.

In this section, we describe how the product pictures can be used to build recognition models for the respective object class. The next section discusses how object information from the product pages can be acquired in an automated way. As a test page, we chose *germandeli.com*, a shop for German products in the United States, because of its well-organized category structure, the clean object pictures, and

the English descriptions of German products which we could easily buy. The system is in no way limited to this specific web site, though sites that are rather based on search than on categories (e.g. *amazon.com*) will provide less structured information.

To make use of the product pictures, we designed and implemented the Objects of Daily Use Finder (*ODUfinder*)[2], an open-source perception system that can deal with the detection of a large number of objects in a reliable and fast manner. Even though it can detect and recognize textured as well as untextured objects, we hereby do not report about the latter. The models for perceiving the objects to be detected and recognized can be acquired autonomously using either the robot's camera or by loading large object catalogs such as the one by *GermanDeli* into the system.

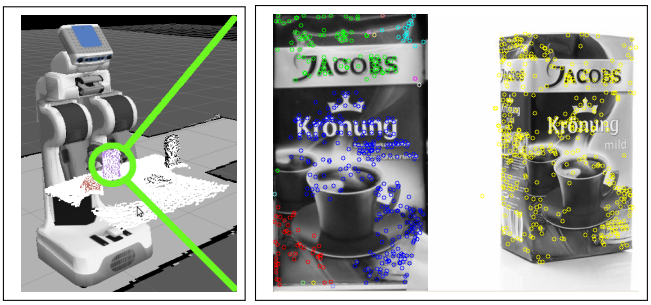


Fig. 10. Left: Region of Interest extraction using cluster segmentation and back-projection of 3D points, Right: Over-segmentation using a region-growing based approach on an object segmented out of the pointcloud shown in the left image.

Product pictures from online shops can provide good models of the texture of objects, but do not contain information about their scale. For manipulation, accurate scaling information is crucial and, in our system, was obtained by combining the 2D image-based recognition with information from a 3D sensor.

For obtaining a 3D pose hypothesis, we use the observation that, in human living environments, objects of daily use are typically standing on horizontal planar surfaces, or as physics-based image interpretation states it, they are in "stable force-dynamic states". The scenes they are part of can either be cluttered, or the objects are isolated in the scene. While the solution of the former is still ongoing work, we solve the latter by a combined 2D-3D extraction of objects standing more or less isolated on planar surfaces.

This combined 2D-3D object detection takes a 3D point cloud, acquired by a tilting laser scanner, and a camera image of the same scene as its inputs. Figure 10 left shows how the system detects major horizontal planar surfaces within the point cloud and segments out point clusters that are supported by these planes [29]. The identified clusters in the point cloud are then back-projected into the captured image to form the region of interest that corresponds to the object candidate.

The *ODUfinder* then employs a novel combination of Scale Invariant Features (SIFT) [15] for textured objects using a vocabulary tree [23], which we extend in two important ways: First, the comparison of object descriptions is done probabilistically instead of relying on the more error-prone original implementation with the accumulation of query sums. Second, the system detects candidates for textured object parts by over-segmenting image regions, and then combines the evidence of the detected candidate parts in order to infer the presence of the complete object (see Figure 10 right). These extensions substantially increase the detection rate as well as the detection reliability, in particular in the presence of occlusions and difficult lighting conditions like specular reflections on object parts. In the current *ODUfinder* configuration, the robot is equipped with an object model library containing about 3500 objects from *Germandeli* and more than 40 objects from the *Semantic3D* database[3]. The system achieves an object detection rate of 10 frames per second and recognizes objects reliably with an accuracy of over 90%. Object detection and recognition is fast enough not to cause delays in the execution of robot tasks.

VIII. Object properties

Often, the way in which a task is performed depends on certain properties of the manipulated objects: Cutlery should be searched for in other places than dairy products, frozen items need to be put into the freezer, and fragile items have to be handled with special care. For manufactured products, such information can be acquired from online shops like *germandeli.com*. We implemented a system that automatically translates the category structure of the website into a subclass structure in the knowledge base: For example, *Dallmayr Prodomo Coffee* is represented as a sub-class of *Dallmayr coffee*, *Coffee (German Brands)*, *Beverages*, and finally *Groceries*. The translation engine and the generated ontology are publicly available as open-source software[4].

In addition to the category structure, online shops also provide detailed descriptions of the properties of products, such as pictures, the perishability status, price, ingredients, etc. Often, this information is already presented in a semi-structured way in form of tables or image icons, so that it can easily and automatically be extracted and added to the knowledge base as properties of the respective object classes. Since the semantic information and the pictures that were used to construct the recognition model originate from the same source, they can easily be combined and allow the robot not only to recognize objects (see Section VII) but also to know their properties and relations to other objects. Otherwise, the problem of integrating knowledge from different sources is much more difficult: Products in an online shop are named differently than in WordNet, Cyc, ehow.com or other sources, and the robot needs to find out if they refer to the same thing.

Obviously, the website parser has to be adapted to different web sites, matching rules or links to existing knowledge need to be added manually, but otherwise, this semi-automatic import generalizes to different information sources and greatly

[2]http://www.ros.org/wiki/objects_of_daily_use_finder

[3]http://ias.cs.tum.edu/download/semantic-3d

[4]http://code.in.tum.de/pubsvn/knowrob/tags/latest/comp_germandeli

speeds up the generation of large knowledge bases. Using only the *germandeli.com* website, we generated an ontology of more than 7,000 object classes including their properties which the robot can both recognize and reason about.

IX. Discussion

The approaches mentioned above are only first steps towards robots that can competently acquire and execute everyday manipulation tasks. In our ongoing research, we are developing mechanisms for (semi-)autonomously acquiring the knowledge required for such tasks. However, there are still many gaps, i.e. pieces of knowledge that we did not find a source for, or sources of information that could not be used completely. Especially understanding longer, complete sentences that are not explicitly written for being understood by robots remains a challenging problem and requires more sophisticated natural language processing techniques than we have used so far. Another problem is the reliable disambiguation of natural language information, taking all available sources of knowledge into account. We expect results from the area of natural language processing to help with these problems.

While the interest in using web information for robotics has increased over the past years, most of the systems use only isolated pieces of information like the set of objects in task instructions [27] or single statements in the OMICS database [26]. This is definitely a good start and shows both the feasibility and usefulness of extracting information from web sources. However, in order to have a robot perform realistic tasks *mainly* based on information on the web, these single pieces have to be put in relation, and the completeness of information becomes an important issue. Therefore, more research is needed to find suitable representations to integrate different sources of knowledge – a task we are using the KnowRob knowledge base for.

We hope that more researchers will start to work on these topics to enable robots to acquire knowledge from the web and to remove one important bottleneck that keeps robots from skilled everyday manipulation: the lack of knowledge. We expect the web to become an important source of knowledge for autonomous robots, though it cannot be the only one. Some kinds of information are hard to find on websites, others depend on the robot's environment or the preferences of the humans it interacts with. So the web knowledge will have to be complemented by information obtained from dialogs with humans, by learning from experience, and by teaching and imitation.

Acknowledgments

This work is supported in part within the DFG research cluster *Cognition for Technical Systems – CoTeSys*, by the EU FP7 Project *RoboEarth* (grant number 248942), and is supported by MVTec GmbH, München.

References

[1] M. Banko, O. Etzioni, and T. Center. The tradeoffs between open and traditional relation extraction. *Proceedings of ACL-08: HLT*, pages 28–36, 2008.

[2] Dave Beckett. RDF/XML syntax specification. Technical report, W3C, 2004.

[3] Michael Beetz. *Plan-based Control of Robotic Agents*, volume LNAI 2554 of *Lecture Notes in Artificial Intelligence*. Springer Publishers, 2002.

[4] Bertram Drost, Markus Ulrich, Nassir Navab, and Slobodan Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010.

[5] C. Fellbaum. *WordNet: an electronic lexical database*. MIT Press USA, 1998.

[6] Rakesh Gupta and Mykel J. Kochenderfer. Common Sense Data Acquisition for Indoor Mobile Robots. In *AAAI*, pages 605–610, 2004.

[7] Ulrich Klank, Dejan Pangercic, Radu Bogdan Rusu, and Michael Beetz. Real-time cad model matching for mobile manipulation and grasping. In *9th IEEE-RAS International Conference on Humanoid Robots*, Paris, France, December 7-10 2009.

[8] Ulrich Klank, Muhammad Zeeshan Zia, and Michael Beetz. 3D Model Selection from an Internet Database for Robotic Vision. In *International Conference on Robotics and Automation (ICRA)*, 2009.

[9] Dan Klein and Christopher D. Manning. Accurate unlexicalized parsing. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 423–430, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[10] Lars Kunze, Moritz Tenorth, and Michael Beetz. Putting People's Common Sense into Knowledge Bases of Household Robots. In *33rd Annual German Conference on Artificial Intelligence (KI 2010)*, Karlsruhe, Germany, September 21-24 2010. Springer.

[11] T.B. Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 284(5):34–43, 2001.

[12] S. Lemaignan, Ros R., Msenlechner L., Alami R., and M. Beetz. Oro, a knowledge management module for cognitive architectures in robotics. In *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.

[13] D.B. Lenat. CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38, 1995.

[14] Hugo Liu and Push Singh. ConceptNet: A Practical Commonsense Reasoning Toolkit. *BT Technology Journal*, 22:211–226, 2004.

[15] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[16] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 1 edition, July 2008.

[17] David Martin, Mark Burstein, Erry Hobbs, Ora Lassila, Drew Mcdermott, Sheila Mcilraith, Srini Narayanan, Bijan Parsia, Terry Payne, Evren Sirin, Naveen Srinivasan, and Katia Sycara. OWL-S: Semantic Markup for Web Services. Technical report, W3C, 2004.

[18] D. McDermott. Robot planning. *AI Magazine*, 13(2):55–79, 1992.

[19] Drew McDermott. The 1998 AI planning systems competition. *AI Magazine*, 21(2):35–55, Summer 2000.

[20] Boris Motik, Peter F. Patel-Schneider, Bijan Parsia, Conrad Bock, Achille Fokoue, Peter Haase, Rinke Hoekstra, Ian Horrocks, Alan Ruttenberg, Uli Sattler, and Mike Smith. OWL 2 web ontology language: Structural specification and functional-style syntax. Technical report, W3C, 2009.

[21] Boris Motik, Rob Shearer, and Ian Horrocks. Hypertableau Reasoning for Description Logics. *Journal of Artificial Intelligence Research*, 36:165–228, 2009.

[22] I. Niles and A. Pease. Towards a standard upper ontology. In *Proceedings of the international conference on Formal Ontology in Information Systems-Volume 2001*, pages 2–9. ACM, 2001.

[23] David Nister and Henrik Stewenius. Scalable recognition with a vocabulary tree. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2161–2168, Washington, DC, USA, 2006. IEEE Computer Society.

[24] M. Paolucci, T. Kawamura, T. Payne, and K. Sycara. Semantic matching of web services capabilities. *The Semantic Web?ISWC 2002*, pages 333–347, 2002.

[25] T. Pedersen, S. Patwardhan, and J. Michelizzi. WordNet:: Similarity: measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004 on XX*, pages 38–41. Association for Computational Linguistics, 2004.

[26] W. Pentney, M. Philipose, J. Bilmes, and H. Kautz. Learning large scale common sense models of everyday life. In *Proceedings of the National Conference on Artificial Intelligence*, volume 22, page 465. Menlo Park, CA, 2007.

[27] Mike Perkowitz, Matthai Philipose, Kenneth Fishkin, and Donald J. Patterson. Mining models of human activities from the web. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 573–582. ACM, 2004.

[28] Eric Prud'hommeaux and Andy Seaborne. Sparql query language for rdf (working draft). Technical report, W3C, 2007.

[29] Radu Bogdan Rusu, Ioan Alexandru Sucan, Brian Gerkey, Sachin Chitta, Michael Beetz, and Lydia E. Kavraki. Real-time Perception-Guided Motion Planning for a Personal Robot. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, St. Louis, MO, USA, October 11-15 2009.

[30] Push Singh, Thomas Lin, Erik T. Mueller, Grace Lim, Travell Perkins, and Wan Li Zhu. Open mind common sense: Knowledge acquisition from the general public. In *CoopIS/DOA/ODBASE*, 2002.

[31] Push Singh and William Williams. LifeNet: A Propositional Model of Ordinary Human Activity. In *Workshop on Distributed and Collaborative Knowledge Capture (DC-KCAP)*, 2003.

[32] Evren Sirin, Bijan Parsia, Bernardo C. Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: A practical OWL-DL reasoner. *Web Semant.*, pages 51–53, 2007.

[33] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 1470–1477, October 2003.

[34] F.M. Suchanek, G. Kasneci, and G. Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM, 2007.

[35] Moritz Tenorth and Michael Beetz. KnowRob — Knowledge Processing for Autonomous Personal Robots. In *IEEE/RSJ International Conference on Intelligent RObots and Systems.*, 2009.

[36] Moritz Tenorth, Daniel Nyga, and Michael Beetz. Understanding and Executing Instructions for Everyday Manipulation Tasks from the World Wide Web. In *IEEE International Conference on Robotics and Automation (ICRA).*, 2010.

[37] Markus Ulrich, Christian Wiedemann, and Carsten Steger. Cad-based recognition of 3d objects in monocular images. In *International Conference on Robotics and Automation*, pages 1191–1198, 2009.

[38] Fei Wu and Daniel S. Weld. Autonomously semantifying wikipedia. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 41–50, New York, NY, USA, 2007. ACM.

[39] Muhammad Zeeshan Zia, Ulrich Klank, and Michael Beetz. Acquisition of a Dense 3D Model Database for Robotic Vision. In *International Conference on Advanced Robotics (ICAR)*, 2009.