Semantic Grasping: Planning Robotic Grasps Functionally Suitable for An Object Manipulation Task

Hao Dang and Peter K. Allen

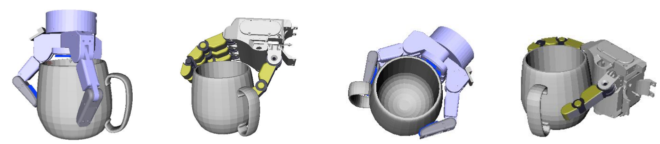
***Abstract*— We design an example based planning framework to generate *semantic grasps*, stable grasps that are functionally suitable for specific object manipulation tasks. We propose to use partial object geometry, tactile contacts, and hand kinematic data as proxies to encode *semantic constraints*, which are task-related constraints. We introduce a *semantic affordance map*, which relates local geometry to a set of predefined semantic grasps that are appropriate to different tasks. Using this map, the pose of a robotic hand can be estimated so that the hand is adjusted to achieve the ideal approach direction required by a particular task. A grasp planner is then used to generate a set of final grasps which have appropriate stability, tactile contacts, and hand kinematics along this approach direction. We show experiments planning semantic grasps on everyday objects and executing these grasps with a physical robot.**

I. Introduction

Grasp planning is a fundamental problem in the field of robotics which has been attracting an increasing number of researchers [1], [2], [3], [4], [5], [6], [7]. Previously proposed methods are reasonably successful in generating stable grasps for execution. However, if we consider planning a grasp for a specific manipulation task, the stability of the grasp is no longer sufficient to describe all of the constraints on the grasp. Such constraints include relative hand orientation, specific object parts the hand should make contact with, or specific regions of the object the hand should avoid. We call these constraints required by a specific task ***semantic constraints***. As we will show in Figure 1 and Section III, a good robotic grasp should satisfy the semantic constraints associated with an intended manipulation task.

In our work, we take an example-based approach to build a grasp planner that considers semantic constraints of specific tasks as a planning criterion and searches for stable grasps satisfying these semantic constraints. This approach is inspired by psychological research which showed that human grasping is to a very large extent guided by previous grasping experience [8]. To mimic this process, we propose that semantic constraints can be embedded into object geometry, tactile contacts and hand kinematics. We design a semantic affordance map which contains a set of depth images from different views and example grasps that satisfy the semantic constraints of different tasks. These depth images help infer the approach direction with respect to the object, guiding the hand to an ideal approach direction. Predefined example grasps provide hand kinematic and tactile information to the planner as references to the ideal hand posture and tactile contact formation. Utilizing this information, the planner searches for stable grasps with an ideal approach direction, hand kinematics, and tactile contact formation.

This work has been funded by NSF Grant IIS-0904514. Both authors are with Department of Computer Science, Columbia University, New York, NY, 10027, USA, {dang,allen}@cs.columbia.edu



(a) (b) (c) (d)

Fig. 1. Stable grasps with different semantic constraints satisfied. All grasps are suitable for *Pick-and-Place* task. However, in the first two grasps, the hand blocks the opening area of the mug. Thus, they are not suitable for *Pouring-Water* task.

II. Related Work

For planning stable robotic grasps, Ciocarlie and Allen proposed the eigen-grasp idea [1]. This method effectively reduces the dimension of the search space for grasp planning and results in a faster search process for form-closure grasps. Based on this approach, a data-driven grasping pipeline is proposed by Goldfeder [9]. Geidenstam approximated 3D shapes with bounding boxes on decomposed objects and trained a neural network to learn good grasps [5]. Saxena [2] and Popovic [4] used synthesized image data to train a classifier to predict grasping points based on features extracted from 2D images. Berenson and Srinivasa proposed a method to generate collision-free stable grasps for dexterous hands in cluttered environments [10].

In addition, there has also been some work in planning grasps considering graspable parts and specific tasks. Researchers, such as Li and Sastry [11], Prats et al. [3], and Haschke et al. [6], analyzed task-oriented grasping using task wrench space. These approaches are mainly based on the analysis of the contacts and the potential wrench space of a grasp. Rosales et al. presented a method to solve the configuration problem of a robotic hand to grasp a given object with a specific contact region [12]. Li et al. took a data-driven approach to grasp synthesis using pre-captured human grasps and task-based pruning [13]. Song et al. designed a Bayesian network to model task constraints in goal-oriented grasping [14]. Using a box-based planning approach, Huebner and Kragic presented a pipeline to generate grasps with some task constraints satisfied [15]. Detry et al. developed a method to analyze grasp affordance on objects based on object-edge reconstructions [16]. Aleotti and Caselli proposed a part-based planning algorithm to generate stable grasps from human demonstration [17]. Sahbani and El-Khoury proposed a method to plan grasps on handles of objects by training a classifier on synthesized data [7]. It is widely accepted that

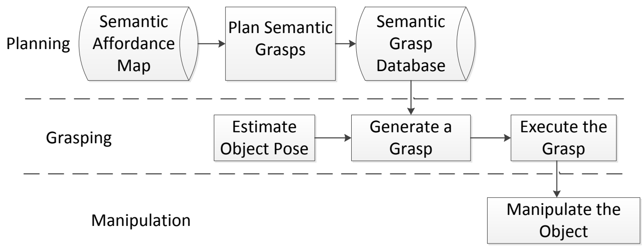


Fig. 2. A typical data driven grasping pipeline considering semantic constraints. In the planning stage, semantic grasps are planned and stored into a semantic grasp database. In a grasping process, planned semantic grasps can be retrieved and executed based on requested manipulation tasks. Once the semantic grasps are executed, a manipulation chain as proposed in [18] could be executed to accomplish a manipulation task.

many everyday objects are designed with a graspable part (e.g. a handle). However, with certain knowledge of where the graspable part is, it is still difficult to determine how to grasp the graspable part appropriately.

In fact, looking solely at the stability of the grasp, the potential task wrench space, or the location of the graspable part of an object leaves out the following important semantic constraints for object grasping in the context of a specific manipulation task. The first constraint is how a robotic hand is oriented and shaped with respect to the object. The second constraint is the locations of the contacts between the hand and the object. In our work, we are trying to encode them directly and use them to plan semantic grasps accordingly.

III. Semantic Grasping

A. *Semantic constraints and data-driven grasping*

Every robotic grasp is used for an intended manipulation task. To perform a specific manipulation task, some constraints are required to be satisfied by the robotic grasp. For a mug, *Pick-and-Place* and *Pouring-Water* are two possible manipulation tasks. For *Pick-and-Place*, stability is one constraint. This constraint requires a grasp to be able to resist possible external disturbances during the manipulation process. For *Pouring-Water*, stability is still necessary, but in addition to this, an extra constraint might require the robotic hand to avoid blocking the opening area of the mug or to grasp the handle of the mug.

In order to plan appropriate grasps for different tasks, satisfying semantic constraints is essential. Figure 1 shows some examples of robotic grasps on a mug that are evaluated as stable grasps according to the epsilon quality [19]. All the grasps in Figure 1 are stable in terms of force/form closure metric and they are all suitable for a *Pick-and-Place* task. However, if we consider using these grasps for a *Pouring-Water* task, only grasps shown in Figure 1(c) and 1(d) are suitable because in the first two grasps the palm blocks the opening area of the mug conflicting with the second semantic constraint required by a *Pouring-Water* task. This example demonstrates that semantic constraints for grasps should be considered in grasp planning procedures.

In our work, we follow the data-driven grasping pipeline where grasp planning and execution are separated and focus on developing a planning method that considers semantic constraints. Figure 2 illustrates a typical data-driven grasping pipeline where our method fits in: compared to traditional data-driven grasping pipelines, our method considers semantic constraints in the planning stage while keep the rest of the pipeline intact. It is worth noting that the planning takes place off-line and is within a simulation environment that is separated from the physical execution.

B. *Embedding semantic constraints*

Semantic constraints are high-level concepts that are difficult to describe and difficult to generalize. Instead of representing semantic constraints explicitly, we attempt to specify semantic constraints using a predefined example grasp and use the example grasp to infer corresponding semantic constraints.

Many everyday objects are designed such that their geometries are appropriate for the corresponding manipulation tasks that they are associated with. For example, a mug has a handle which is designed to be grasped. For a *Pouring-Water* task, it is always a good strategy to grasp the body of the mug or to grasp the handle from the direction in which it stretches away from the body, because these two grasps satisfy the two semantic constraints a *Pouring-Water* task requires: 1) grasp stability and 2) avoid blocking the opening area of the mug.

Semantic constraints imply requirements on the following aspects: 1) part of the object to be grasped, 2) relative orientation of the hand to the graspable part, 3) hand posture, and 4) contact locations on the hand.

The graspable part of an object being approached by a robotic hand can be encoded using 3D depth images of the object from the approach direction of the hand. The depth images describe the partial geometry in view. It also indicates the orientation of this part of the object with respect to the hand. Hand posture can be derived directly from the joint values of the hand and contact information can be extracted from a set of tactile sensor arrays on the hand. Thus, we propose embedding semantic constraints into these related sensory data. Given an example grasp which has already satisfied specific semantic constraints, we can compare these quantities to those of candidate grasps on the same object or novel objects of the same class. If they are similar, we consider the corresponding semantic constraints as satisfied. Otherwise, we consider the semantic constraints as unsatisfied.

C. *Semantic Grasp: a definition*

We define the ***semantics*** of a robotic grasp as the intended manipulation task whose semantic constraints are satisfied by the grasp. It is a symbolic label or a phrase that uniquely identifies a task of the object (e.g. *Pouring-Water*).

A ***semantic grasp*** is a robotic grasp that satisfies the semantic constraints imposed by a manipulation task. We write a semantic grasp formally as

$$\mathcal{SG} = < \mathcal{S}, \mathcal{T}, \mathcal{K} >$$

where $\mathcal{S}$ is the semantic label of the grasp, e.g. *Pouring-Water*; $\mathcal{T}$ is the tactile contacts of the grasp, e.g. arrays of tactile sensor reading; $\mathcal{K}$ is the hand kinematic data of the grasp, e.g. a set of joint angles and the orientation and location of the wrist.

D. Semantic Affordance Map: a definition

In Section III-B, we discussed that the semantic constraints can be indirectly embedded in the object range data, hand kinematic data, and tactile contacts. We now introduce a semantic affordance map $\mathcal{M}_{\mathcal{C}}$ to associate semantic grasps with an object class $\mathcal{C}$.

A ***semantic affordance map*** $\mathcal{M}_{\mathcal{C}}$ is a set of triples:

$$\mathcal{M}_{\mathcal{C}} = \{< \mathcal{P}, \mathcal{F}(\mathcal{D}), \mathcal{B} >\}, \quad (\mathcal{B} = \{\mathcal{SG}\})$$

where $\mathcal{P}$ is the approach direction from the hand to the object; $\mathcal{D}$ is a depth image of the object from $\mathcal{P}$; $\mathcal{F}(\cdot)$ is a function that extracts features from $\mathcal{D}$; $\mathcal{B}$ is a set of semantic grasps from $\mathcal{P}$.

In this map, the following two relations are established.

1. $\mathcal{F}(\mathcal{D}) \rightarrow \mathcal{P}$, given an image feature descriptor $\mathcal{F}(\mathcal{D})$ of a depth image of the object from a particular viewing angle (approach direction), this mapping tells us the current approach direction of the hand to the object. If the object is symmetric, this mapping can be one to many.

2. $\mathcal{F}(\mathcal{D}) \rightarrow \mathcal{B}$, given an image feature descriptor $\mathcal{F}(\mathcal{D})$ of a depth image of the object from a particular viewpoint, this mapping tells us the possible semantic grasps on the corresponding geometry.

A semantic affordance map is considered as a manual for semantic usage of an object. In a semantic affordance map, it is probable that many triples have an empty set of semantic grasps. This is because there are many approach directions that are not good for any manipulation tasks. So, only a few $\mathcal{B}$'s in a semantic affordance map contain semantic grasps.

E. Method overview: planning semantic grasps

Our planning method is inspired by Castiello who showed that both cognitive cues and knowledge from previous experience play major roles in visually guided grasping [8]. We use an example-based approach to mimic this experience-based method. By analogy, $\mathcal{M}_{\mathcal{C}}$ acts like an experience base. $\mathcal{B}$ records all the successful grasps that are experienced before. $\mathcal{P}$ and $\mathcal{D}$ are used to mimic human knowledge of the object geometry.

To plan a grasp with semantics $\mathcal{S}$ on an object of class $\mathcal{C}$, we assume a semantic affordance map on this object class, $\mathcal{M}_{\mathcal{C}}$, has been given. First, a semantic grasp of semantics $\mathcal{S}$ is retrieved from $\mathcal{M}_{\mathcal{C}}$. Then, an initial approach direction is randomly chosen and a depth image of the object from this approach direction is taken. With the depth image, the current hand approach direction to the object is estimated by looking up in the semantic affordance map. Utilizing this estimated approach direction, along with the tactile and kinematic information stored in the predefined semantic grasp, our method adjusts the hand to the ideal approach direction and searches along the ideal approach direction for grasps that have similar tactile contact formation and hand kinematics.

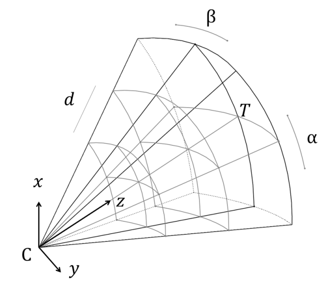


Fig. 3. Shape context computation. $C - xyz$ is the camera coordinate system with its origin at point C. In an eye-on-hand system, C is also the origin of the palm. The three dimensions of the spherical coordinate system for our shape context feature are as follows: d is the depth from the origin of the camera; α is the latitude in the camera coordinate system; β is the longitude in the camera coordinate system.

With all these similarities being achieved, we consider the semantic constraints specified by the predefined semantic grasp are also satisfied by the newly planned grasp. In the following sections, we discuss in detail how to build a semantic affordance map for an object class and how a grasp planner uses it to plan semantic grasps for a specific manipulation task.

IV. Building A Semantic Affordance Map

A semantic affordance map is dedicated to one object class. It is built in simulation and is designed to be a knowledge base that stores semantic grasps suitable for possible manipulation tasks an object could be involved in. It also provides hints to a grasp planner about how to satisfy these semantic constraints by relating semantic grasps with partial object geometry and approach directions. To build a semantic affordance map, we first choose a representative object of an object class and collect depth images from different approach directions to this object. Then, these depth images are encoded and example semantic grasps are manually defined.

A. Sampling strategy

To sample around an object in simulation and obtain the object's partial geometry information, we first create a unit sphere centered at the geometrical center of the object model. Along the latitude lines and the longitude lines, every two degrees, we collect a depth image of the object model using OpenGL. The virtual camera in OpenGL is placed at the crossing of the longitude and latitude with its horizontal axis aligned with the latitude line and its depth axis aligned with the radius of the sphere. We also move the virtual camera along the radius such that the bounding box of the object model is right within the field of view. By doing this, we make sure that the entire object model is in the view.

B. Encoding a depth image

Given a set of sampled depth images using the strategy above, we encode them such that they can be used as keys to index all the samples effectively. Then, using the relation $\mathcal{F}(\mathcal{D}) \rightarrow \mathcal{P}$ as in Section III-D, we can estimate the approach direction of the hand given a depth image from an unknown direction. To encode a depth image, we use a similar idea from the shape context method [20].

Figure 3 illustrates the spherical coordinate system to compute the shape context, which co-locates with the camera's coordinate system $C - xyz$. The discretization of the spherical space is along the latitude (α), the longitude (β), and the depth (d). In this case, the shape context vector stores the distribution of the object surface points over this discretized spherical segment where the points exist.

C. Recording semantic grasps

From the previous sampling step, we have already computed all the $\mathcal{P}$'s and $\mathcal{F}(\mathcal{D})$'s of a semantic affordance map. To input sets of semantic grasps and complete a semantic affordance map, we manually select a good approach direction for each possible manipulation task with some semantics. Along this approach direction, a semantic grasp can be specified manually. Then, the following information is recorded and associated with this semantic grasp.

1) Hand kinematics: Hand kinematic data is stored only with a semantic grasp. This data indicates the shape of the hand when the grasp is applied. To store the hand kinematics, we store the angle value for each joint.

2) Tactile contacts: In order to represent where a hand is in contact with the object, we take into consideration the tactile contacts when a grasp is being applied on an object. In a semantic grasp, we use $\mathcal{T}$ to store the readings from the simulated tactile sensor pads.

If there is more than one approach direction for a specific manipulation task, we apply the same method to define additional semantic grasps along other approach directions.

V. Planning Semantic Grasps

Given a manipulation task and a semantic affordance map, planning a semantic grasp on a target object can be thought of as a search for grasps that satisfy the semantic constraints indicated by a predefined semantic grasp. A planning procedure includes the following three steps.

A. Step 1: Retrieve semantic grasps

The first step is to retrieve a predefined semantic grasp from the semantic affordance map. This is done by searching within the semantic affordance map and looking for semantic grasps with an appropriate semantic label according to the requested manipulation task. In the following steps of the pipeline, we will use this semantic grasp as a reference for planning.

B. Step 2: Achieve the ideal approach direction

The first semantic constraint we need to satisfy is the geometric constraint which requires a specific part of the object to be grasped. This constraint can be implicitly inferred by the approach direction of the hand. So, in order to get to the most appropriate approach direction required by the semantic constraints, we first estimate the current approach direction of the hand. To do this, a depth image of the object is taken from the hand's current approach direction. We encode the depth image as in Section IV-B to get the shape context feature. Then, we look up in the semantic affordance map and search for k nearest neighbors based on this geometry feature.

To match against the entries in the semantic affordance map, we used χ^2 distance to calculate the difference between two geometry features. Since k could be larger than one, we need to use some rules to decide a neighbor that is most widely agreed among these k nearest neighbors. To do this, we calculate a cumulative distance for each neighbor from the remaining neighbors, which indicates the extent to which other neighbors disagree with it. Algorithm 1 illustrates this scheme, where $D(\cdot)$ denotes a distance function that calculates the actual angle between the approach directions represented by the two neighbors.

Algorithm 1: Computing the most agreed neighbor

Input: k nearest neighbors $N = \{n_1, ..., n_k\}$
Output: the most agreed neighbor n_m
Initialize array v with k entries for cumulative distances
foreach $n_i \in N$ **do**
    foreach $n_j \in N - \{n_j\}$ **do**
        $v[i] += D(n_i, n_j)$;
        $v[j] += D(n_i, n_j)$;
    end
end
n_m = 1^{st} neighbor with a minimum cumulative distance
return n_m

Once the current approach direction is estimated, adjustment can be done by calculating the transform between the current approach direction and the ideal approach direction that satisfies the semantic constraints.

C. Step 3: Refine the grasp

Based on the previous two steps, a promising hand approach direction has been achieved for the specific manipulation task. This is only a good start to satisfy all the semantic constraints embedded in the predefined semantic grasp, because solely relying on the approach direction is not sufficient. For example, the stability of the grasp, similar kinematics, and tactile contacts of the hand cannot be guaranteed simply by approaching the object from an ideal direction. We consider them in a grasp refinement step. In this step, we first use the eigen-grasp planner to generate a sequence of potential stable grasps along the approach direction [1]. We then sort them according to their similarities with the predefined semantic grasp.

1) Grasp Stability: To ensure the grasp stability, we use epsilon quality as a quality metric [19]. The planner checks the epsilon quality each time it finds a promising solution and outputs only those grasps that have positive epsilon qualities.

2) Hand Kinematics: During this refinement procedure, we use an eigen-grasp planner to search for grasps around the ideal approach direction as discussed in [1]. We first place a constraint on the hand kinematics, i.e. the DOF's

TABLE I

OBJECTS, MANIPULATION TASKS, AND SEMANTIC CONSTRAINTS

Object	Manipulation Task	Semantic Constraints
Mug	pour water	not blocking the opening area
Phone	answer a call	grasping the handle
Door handle	pull/push the door	power grasping the mid-point
Door handle	slide the door	hook grasping the mid-point
Drill	hold and drill	grasping the handle

of the hand. We use the corresponding eigen-grasps of the DOF values recorded in the semantic grasp as a kinematic reference and probabilistically limit the search region to be around this kinematic reference. Based on this, the output grasps of the planner should share similar kinematics of the example semantic grasp, maximumly preserving a similar hand posture during grasping.

3) Tactile Contacts: Similar tactile contact formation as recorded in the predefined semantic grasps is achieved by comparing the reading of each sensor pad attached to each link. With similar tactile contacts, we expect the hand to touch similar parts of the object which improves the possibility that the planned grasp holds the object in the way which is defined in the example semantic grasp.

VI. EXPERIMENTS

We conducted two levels of experiments: planning semantic grasps inside the *GraspIt!* simulator [21] and grasping physical objects with planned semantic grasps using a physical robot, which correspond to the first two levels shown in Figure 2. Table I summarizes everyday objects that were chosen for our test and the corresponding semantic constraints of each task the grasp should be suitable for.

A. Planning semantic grasps

In table-top object manipulation tasks, objects are usually placed in their canonical upright orientations. So, we assumed that all the object models were defined in their canonical upright orientations with respect to a common plane. For each of the object classes in Table I, we chose a source object model as a representative (shown in the left part of the third column in Figure 4) to build a semantic affordance map. Using OpenGL, we first generated depth images of the representative object model from different approach directions. To define example semantic grasps, we used the *Graspit!* simulator to manually record stable grasps suitable for related tasks. Tactile sensors were simulated based on the soft finger model proposed by Ciocarlie et al. [22]. By associating example semantic grasps with the depth images, we built a semantic affordance map for an object class. With a semantic affordance map, we then chose different object models from the same object class as targets and used the proposed algorithm to plan semantic grasps on them. The target objects were different from the source models that were used to build the semantic affordance maps, as they had different sizes and shapes.

In Figure 4, we show experimental results of planning semantic grasps on different object models. In each row, the second column shows a predefined semantic grasp that was stored in the semantic affordance maps. The third column shows the comparison of the geometry between the source(left) and the target(right) objects. The source objects are those ones that were used for building semantic affordance maps. They are different from the target objects, but similar in shape. The last two columns show the top two ranked grasps generated by our planning method according to their tactile and hand posture similarities. The experimental results indicate that, by using our planning algorithm, semantic grasps can be synthesized from similar objects with predefined example semantic grasps.

B. Semantic grasping with a physical robot

Following the planning experiments, we connected our planning method to a grasp execution system and tested an entire grasping pipeline from modeling a physical object using an off-the-shelf 3D scanner to planning a semantic grasp on the model and to executing a semantic grasp for a requested manipulation task on a physical object.

We chose a mug, a phone, and a drill as target objects, shown in experiments 3, 5, and 8 in Figure 4 respectively. A NextEngine 3D scanner was used to obtain geometrical models of the physical objects (shown in the right part of the third column of each experiment). Using our proposed method, we then planned semantic grasps and stored them in a semantic grasp database. In the grasping stage, a target object was placed in front of the robot. A Kinect sensor acquired a 3D point cloud of the scene. The recognition method proposed by Papazov et al. [23] was used in our perception system, which uses partial geometry of an object to recover its full 6D pose. Once the pose of an object was recovered, a planned semantic grasp was retrieved from the semantic grasp database according to the object name and the semantic label. Finally, the OpenRave [24] planner generated a collision-free trajectory to the final grasping pose and the hand moved to the target pose and executed the grasp.

Figure 5 shows snapshots of the process of the grasping pipeline using semantic grasps in the experiments. The first two columns show the predefined semantic grasps on the source objects and the generated semantic grasps on the target objects. The third column shows the physical objects placed in the robot workspace. The fourth column shows the point clouds of the workspace reconstructed from a Kinect sensor. The fifth column shows the final grasp of a Barrett hand.

VII. CONCLUSION AND DISCUSSION

In this paper, we develop an example-based grasp planning method to plan stable robotic grasps which satisfy semantic constraints required by a specific manipulation task. We propose using partial object geometry, hand kinematic data, and tactile contacts to embed semantic constraints. We also introduce a semantic affordance map which relates partial geometry features to semantic grasps. Using this map, our

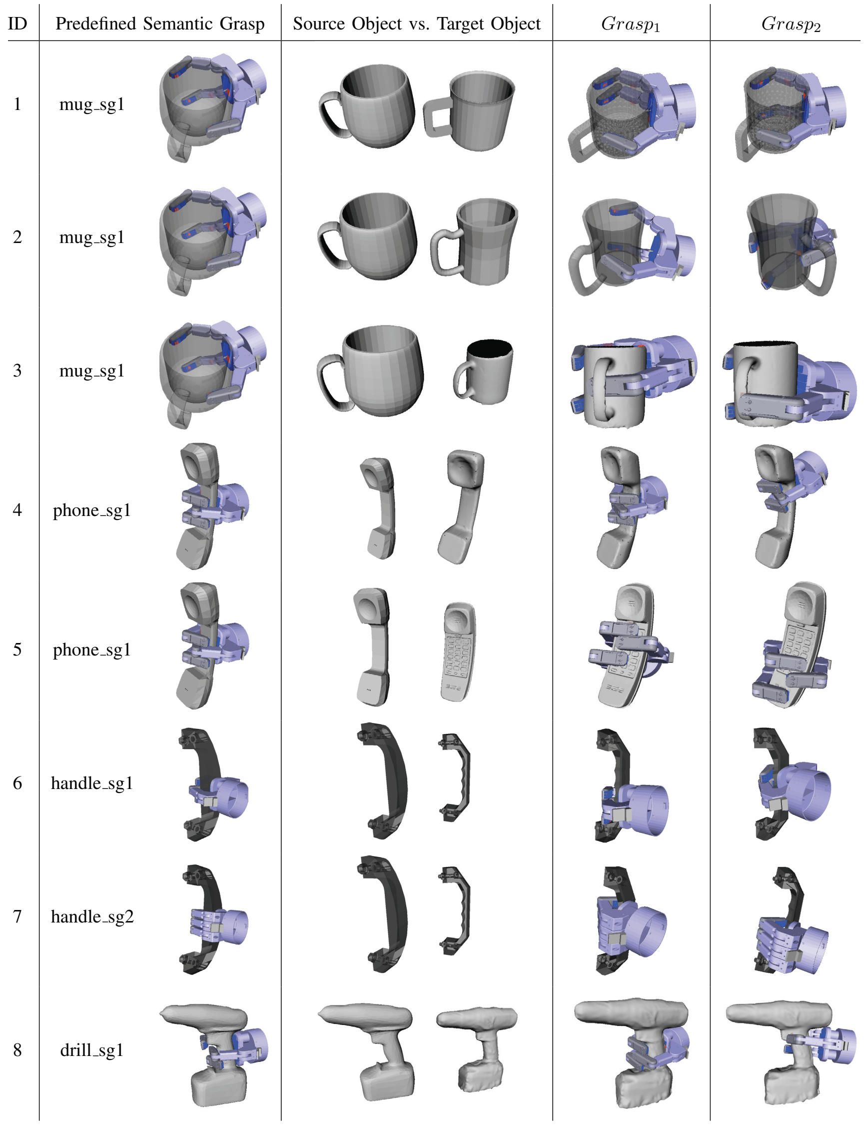


Fig. 4. Semantic grasps planned on typical everyday objects. From left to right: experiment ID, the predefined semantic grasps stored in the semantic affordance map, a pair of source object and target object for each experiment, and top two grasps generated. Last two columns for the top two grasps were obtained within 180 seconds and are both stable in terms of their epsilon quality. Some objects are displayed with transparency to show the grasp.

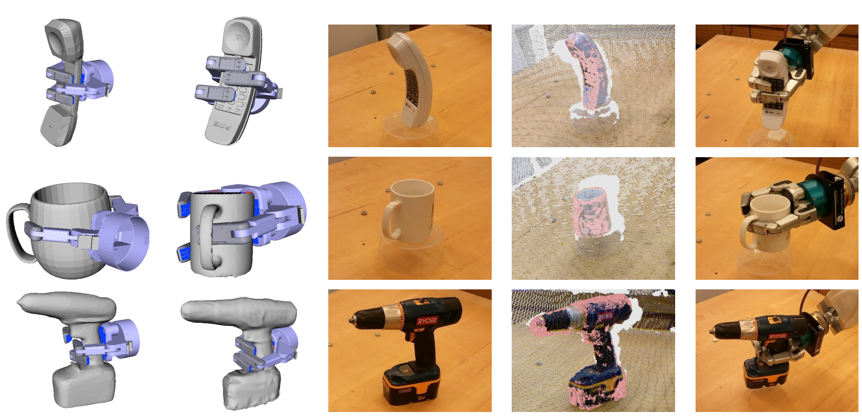


Fig. 5. Grasping physical objects with semantic grasps. From left to right: predefined semantic grasps on source objects, semantic grasps on target objects generated using the proposed method, a snapshot of the physical object in the workspace, reconstructed point cloud of the object in the workspace, and the final grasp of a Barrett hand attached to a Staubli arm. Pink points in the point clouds are object models placed with the estimated pose.

method considers the semantic constraints imposed by a specific task and plans semantic grasps accordingly. We show experiments of planning and executing semantic grasps on everyday objects.

A model of the object may not be necessary. For a system where grasp planning and executing are separated, it is ideal to get the model of the object beforehand. If the object model is not obtained in advance, some existent modeling algorithms can be used to reconstruct the object from the real scene using depth sensors, such as a Kinect sensor. Another approach can be to connect this algorithm with a physical robot system, obtaining depth images directly from physical depth sensors and making hand movements with the robot. In this case, we are merging the virtual world and the physical world. The planning process which used to be in a virtual world now becomes an exploration process in the physical world, defining a control process to achieve a semantic grasp specified in a predefined example grasp, making this planning algorithm more like a control algorithm.

Currently, the refinement step in our method consists of 1) planning a sequence of stable grasps using a stochastic grasp planner and 2) sorting planned grasps based on their similarities to predefined semantic grasps. However, stochastic planning may not be the ideal solution. It can take more time than necessary to find an appropriate grasp. One potential alternative approach is to use local geometry information to synthesize hand adjustment as we proposed in [25].

Symmetric objects may raise challenges for our matching method which estimates the approach direction of the hand since multiple views of a symmetric object could have the same depth images. We believe that by utilizing more sophisticated localization methods this problem could be alleviated.

In our work, we do not consider the kinematic feasibility of a grasp. For example, a planned semantic grasp may not be kinematically feasible in a real environment due to collisions or other workspace constraints. This could be solved by using collision checking to filter out infeasible grasps after a number of good semantic grasps are produced or utilizing algorithms such as in [26] to achieve the required pre-grasp pose of the object.

For the next step, we will be considering possible ways to generalize the semantic affordance map so that it would be easier to transfer grasping knowledge between objects and tasks while preserving their semantic affordance. In our current approach, a semantic affordance map is associated with a specific class of objects and it requires similarities between objects. This, to some extent, limits the application of our approach, especially in an environment which has many different classes of objects that are frequently changing over time. We believe that, in these situations, a more generic representation of semantic affordance map could help.

VIII. Acknowledgements

The authors would like to thank Jonathan Weisz and Lixing Dong in the pursuit of this work.

References

[1] M. T. Ciocarlie and P. K. Allen, "Hand Posture Subspaces for Dexterous Robotic Grasping," *The International Journal of Robotics Research*, vol. 28, no. 7, pp. 851–867, 2009.

[2] A. Saxena, J. Driemeyer, J. Kearns, and A. Y. Ng, "Robotic grasping of novel objects," in *Advances in Neural Information Processing Systems 19*, pp. 1209–1216, 2007.

[3] M. Prats, P. Sanz, and A. del Pobil, "Task-oriented grasping using hand preshapes and task frames," in *Robotics and Automation, 2007 IEEE International Conference on*, pp. 1794 –1799, april 2007.

[4] M. Popovic, D. Kraft, L. Bodenhagen, E. Baseski, N. Pugeault, D. Kragic, T. Asfour, and N. Krüger, "A strategy for grasping unknown objects based on co-planarity and colour information," *Robotics and Autonomous Systems*, vol. 58, no. 5, pp. 551 – 565, 2010.

[5] S. Geidenstam, K. Huebner, D. Banksell, and D. Kragic, "Learning of 2D grasping strategies from box-based 3D object approximations," in *Proceedings of Robotics: Science and Systems*, June 2009.

[6] R. Haschke, J. J. Steil, I. Steuwer, and H. Ritter, "Task-oriented quality measures for dextrous grasping," in *Proc. Conference on Computational Intelligence in Robotics and Automation*, 2005.

[7] A. Sahbani and S. El-Khoury, "A hybrid approach for grasping 3d objects," in *IROS*, pp. 1272 –1277, 2009.

[8] U. Castiello, "The neuroscience of grasping," *Nature Reviews Neuroscience*, vol. 6, pp. 726–736, September 2005.

[9] C. Goldfeder and P. K. Allen, "Data-driven grasping," *Auton. Robots*, vol. 31, pp. 1–20, July 2011.

[10] D. Berenson and S. Srinivasa, "Grasp synthesis in cluttered environments for dexterous hands," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids08)*, December 2008.

[11] Z. Li and S. Sastry, "Task oriented optimal grasping by multifingered robot hands," in *ICRA*, pp. 389–394, 1987.

[12] C. Rosales, L. Ros, J. M. Porta, and R. Suarez, "Synthesizing Grasp Configurations with Specified Contact Regions," *The International Journal of Robotics Research*, July 2010.

[13] L. Ying, J. Fu, and N. Pollard, "Data-driven grasp synthesis using shape matching and task-based pruning," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 13, no. 4, pp. 732 –747, 2007.

[14] D. Song, K. Huebner, V. Kyrki, and D. Kragic, "Learning task constraints for robot grasping using graphical models," in *IROS*, pp. 1579 –1585, 2010.

[15] K. Huebner and D. Kragic, "Grasping by parts: Robot grasp generation from 3d box primitives," in *International Conference on Cognitive Systems*, January 2010.

[16] R. Detry, D. Kraft, A. Buch, N. Kruger, and J. Piater, "Refining grasp affordance models by experience," in *Robotics and Automation, 2010 IEEE International Conference on*, pp. 2287 –2293, may 2010.

[17] J. Aleotti and S. Caselli, "Part-based robot grasp planning from human demonstration," in *ICRA*, pp. 4554 –4560, may 2011.

[18] H. Dang and P. Allen, "Robot learning of everyday object manipulations via human demonstration," in *IROS*, pp. 1284 –1289, oct. 2010.

[19] C. Ferrari and J. Canny, "Planning optimal grasps," in *ICRA*, pp. 2290 –2295 vol.3, May 1992.

[20] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 509–522, 2002.

[21] A. T. Miller and P. K. Allen, "Graspit! a versatile simulator for robotic grasping," *Robotics & Automation Magazine, IEEE*, vol. 11, no. 4, pp. 110–122, 2004.

[22] M. Ciocarlie, C. Lackner, and P. Allen, "Soft finger model with adaptive contact geometry for grasping and manipulation tasks," *World Haptics Conference*, pp. 219–224, 2007.

[23] C. Papazov and D. Burschka, "An efficient ransac for 3d object recognition in noisy and occluded scenes," in *ACCV*, pp. 135–148, 2010.

[24] R. Diankov and J. Kuffner, "Openrave: A planning architecture for autonomous robotics," Tech. Rep. CMU-RI-TR-08-34, Robotics Institute, Pittsburgh, PA, July 2008.

[25] H. Dang and P. K. Allen, "Tactile experience-based robotic grasping," in *Workshop on Advances in Tactile Sensing and Touch based Human-Robot Interaction, HRI*, March 2012.

[26] L. Y. Chang, G. Zeglin, and N. Pollard, "Preparatory object rotation as a human-inspired grasping strategy," in *IEEE-RAS International Conference on Humanoid Robots*, pp. 527–534, December 2008.