Earned the IROS2017 Best Paper Award on Cognitive Robotics

Task-oriented Grasping with Semantic and Geometric Scene Understanding

Renaud Detry Jeremie Papon Larry Matthies
Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA

***Abstract*— We present a task-oriented grasp model, that encodes grasps that are configurationally compatible with a given task. For instance, if the task is to pour liquid from a container, the model encodes grasps that leave the opening of the container unobstructed. The model consists of two independent agents: First, a geometric grasp model that computes, from a depth image, a distribution of 6D grasp poses for which the shape of the gripper matches the shape of the underlying surface. The model relies on a dictionary of geometric object parts annotated with workable gripper poses and preshape parameters. It is learned from experience via kinesthetic teaching. The second agent is a CNN-based semantic model that identifies grasp-suitable regions in a depth image, i.e., regions where a grasp will not impede the execution of the task. The semantic model allows us to encode relationships such as *"grasp from the handle."* A key element of this work is to use a deep network to integrate contextual task cues, and defer the structured-output problem of gripper pose computation to an explicit (learned) geometric model. Jointly, these two models generate grasps that are mechanically fit, and that grip on the object in a way that enables the intended task.**

I. INTRODUCTION

Humans are ingenious: If we cannot find the tool we ordinarily use to perform a task, we easily find another tool that qualifies. This skill is crucial to our ability to handle the large variety of objects that populate our world. Unfortunately, this skill is not yet accessible to today's robots – most factory robotic workers only ever perform a single task, with a single tool. Providing robots with the capability to use new tools and objects is vital to their transition to uncontrolled environments. In this domain, our community has focused on two important issues: developing grasp models and developing task models. Grasp models [16], [27] determine grasping points that are suitable for picking up an object, while task models [15] assume the pre-existence of a satisfactory grip on the object and focus on modeling the motion that realizes the task. The objective of this work is to bridge the gap between these two domains, i.e., grasping objects to the end of completing a task that imposes constraints on the grip configuration. We are investigating a model that allows a robot to grasp a previously-unseen object in a way that is compatible with the execution of a given task.

A straightforward approach to allowing robots to work with new objects is to hardcode the manipulation parameters of each combination of task and object that the robot is susceptible to encounter. While this approach works satisfactorily in a controlled environment where the number of tasks and objects is limited, it does not translate to open-ended environments containing millions of objects, and where each object can be involved in many tasks. To address this problem, we must provide robots with means of transferring task-specific grasp experience across objects, to allow them to initiate a task with objects that are not necessarily those they trained with.

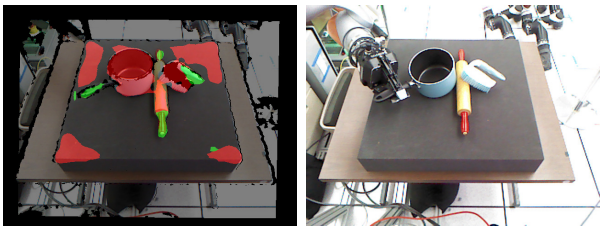


Fig. 1. Grasping for a *transport* task. Left: the green overlay indicates task-compatible regions encoded by the CNN. Right: Planning and executing a 7DOF grasp (pose and preshape) within a compatible region.

The problem of task-oriented grasping intuitively factorizes into two subproblems: (A) decide which areas of an object can be grasped to perform the task, and (B) position the wrist and the fingers around such areas to form a mechanically stable grasp. Problem A relates to the agent's understanding of the world – for instance a semantic understanding of objects, tools and context, or a working knowledge of physics. In turn this understanding allows the agent to formulate task constraints that incite the proper execution of the task. Problem B relates to sensorimotor programs that map from visually-perceived object shapes, to wrist and finger configurations that yield a workable mechanical bond between the gripper and the object. The problem of task-oriented grasping then amounts to jointly optimizing A and B.

Evidence collected in different behavioral and neurophysiological studies [9], [20], [24] indicates that the A–B composition above is possibly consistent with primate grasping. The need for contextual information expressed in Problem A is critically supported by the work of Creem et al. [5], who observed that grasping relied on semantic understanding and noted that *"without semantic processing, the visuomotor system can direct the effective grasp of an object, but not in a manner that is appropriate for its use."*.

The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. This research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-10-2-0016. © 2017 California Institute of Technology. Government sponsorship acknowledged.

Reciprocally, this A–B composition translates into a natural grasp planning solution for artificial agents [1], [6], [14], [31], [33]. Capturing task constraints (Problem A) has been tackled with physics-based simulations [6], visual features and learned statistical models [14], [19], [31], or explicit semantic rules or ontologies [36]. Capturing sensorimotor correlations (Problem B) is addressable with tools readily available in the machine-learning literature, and previous work demonstrating the feasibility of this process is abundant in the robotics community. Authors have for instance demonstrated how feature classification [28], prototypical parts [7], or CNNs [12], [18] are applicable to solving at least in part the problem of geometric grasp planning.

In this paper, we present a model of task-oriented grasping that combines geometric reasoning to semantic scene understanding. We represented task constraints (Problem A) with a CNN that identifies task-compatible areas within images provided by the robot's camera. We trained the CNN on a synthetic, hand-annotated dataset. We constructed this dataset by annotating 3D object meshes with task constraints, and generating random views of random configurations of those objects. This process allowed us to produce a large training dataset while keeping the annotation effort reasonable.

We addressed wrist/finger pose planning (Problem B) with a geometric part-based model that finds object parts whose shape is compatible with the gripper's. To verify task constraints, we restricted the geometric planner to object surfaces that are noted task-compatible by the semantic model. Fig. 1 illustrates our approach.

Our contributions are as follows: We contribute an original solution to task-oriented grasping, that addresses geometric and semantic planning jointly. We demonstrate the applicability of a MultiNet-based CNN architecture for modeling task constraints in image space, and we provide a model that combines geometric and semantic information probabilistically. Our model allows the agent to grasp new objects for which there is no mesh model. It is applicable to 2.5D object images such as those captured by stock RGBD sensors. It is capable of generalizing across objects that are globally different in shape: the geometric planner only exploits local 3D structure, and the CNN learns class traits that are not necessarily anchored in global object structure.

II. RELATED WORK

This work falls under the broad umbrella of *robot affordances* [26], [33], whereby roboticists took inspiration in J. Gibson's ecological perception [10] to formalize robot-world interactions. Sahin et al. [26] provided an extended discussion of Gibson's affordance concept, its adoption by the robotics community, and its application to traversability management (pushing/running through/driving around). Stoytchev [33] studied the use of tools in artificial agents. Similar to this work, Stoytchev made a distinction between *binding affordances* that relate to hand-object bonds, and *output affordances* that encode the additional capability that an agent earns by seizing a tool. His work [33] focused on autonomous acquisition of output affordances via exploratory learning. *Affordance recognition* is of particular relevance to our work, and it has been shown that object affordances can be modeled using hard-coded features [21] or CNNs [25], [32].

The work presented in this paper contributes to the effort supported by the authors listed above, but it focuses specifically on grasp synthesis. By contrast to the work listed above, we assume that the agent has already identified that it can use an object or a set of objects to execute a given task, and we focus on deciding where and how to grasp. Close in spirit to our work, Antanas et al. [1] presented a model that joined task-level and action-level reasoning. The authors first segmented objects into *top*, *middle*, *bottom*, and *handle*, and used a task-dependent grasp model to select a part to grasp. Then, a shape-based grasp model selected the final grasp configuration. We go further by avoiding a hardcoded segmentation into a set of predefined parts, and instead learn a mapping from image data to both semantic constraints and gripper parameters. Song et al. [30] presented a generative model of task-oriented grasping, Dang and Allen [6] presented a model for task-oriented grasping that exploited both visual and tactile information, and Vahrenkamp et al. [35] decomposed objects into parts and assigned task labels to individual parts. While those models have the ability to transfer parameters across objects of similar shape, their ability to transfer to an object whose shape matches none of the training objects is unclear. Transfer to new objects is one of the key features of our model: both components of our model (CNN and geometric planner) rely on local information and would allow us to transfer parameters from a jug to a suitcase if both objects exhibited a similar handle. The work of Hjelm et al. [14] and Nguyen et al. [22] are closest in spirit to our own. Hjelm et al. [14] hardcoded a bank of visual features such as *close to an opening* or *fraction of object surface coverage*, then learned qualitative and geometric relations between those features and grasping points that are compatible with a task. This paper goes beyond the work of Hjelm et al. by learning task features from annotated data. Nguyen et al. [22] trained a neural network on the affordance dataset of Myers et al. [21]. Our work goes beyond Nguyen et al. by using a geometric grasp planner that outputs grasps that are not necessarily centered on a part's centroid, and by demonstrating applicability on dataset designed for task-oriented manipulation, where objects may overlap. Our work complements the *task space regions* of Berenson et al. [2], by providing means of parametrizing task and grasp constraints from vision data.

III. METHOD

Our aim is to define a task-oriented grasp model, that encodes grasps whose placement on an object enable a given task. For instance, if the task is to hand over an object to an operator, the model encodes grasps that leave part of the object's surface available for the operator to secure his own grip. As alluded above, the model consists of two components, a geometric model and a semantic model. The

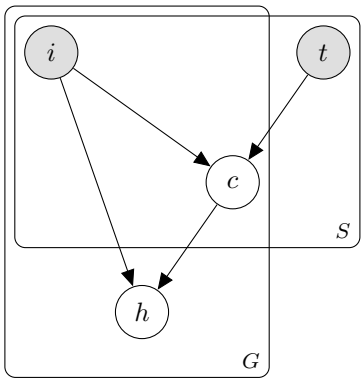


Fig. 2. Graphical representation of our grasp model. Variables i and t model a depth image and the task. Variable c models contact regions (in image space) that are compatible with the task. Variable h models hand poses that comply with the geometry of the scene and with task constraints.

geometric model computes, from a depth image, a distribution of 6D grasp poses for which the shape of the gripper matches the shape of the underlying surface. The model relies on a dictionary of geometric object parts annotated with workable gripper poses and preshape parameters. This model builds on the work of Detry et al. [7], whereby an artificial agent learns such a dictionary from experience via kinesthetic teaching. The second component is a semantic model that encodes task-compatible grasping regions. It relies on a CNN that parses a scene into a set of task-compatible regions, building on the work of Papon et al. [23]. The semantic model allows us to encode relationships such as "grasp from the handle". The product of the geometric and semantic agents allows us to initiate manipulative tasks on previously-unseen objects by identifying grasping regions where the shape of the gripper fits the shape of the tool or object, and where the positioning of the gripper allows the robot to perform the intended task. This work advances the state-of-the art by leveraging data-driven semantic scene understanding and combining a qualitative semantic map to explicit geometric constraints, thereby providing solutions that are both contextually relevant and mechanically realizable.

A. Grasp Model

Our objective is to model the probability that a grasp $h \in \mathcal{H}$ is suitable to accomplish a task t, using an object shown in an image i, where $\mathcal{H}$ models the space of hand configurations (wrist pose and hand preshape), i models a depth image, and t models the task. This objective translates into the construction of an empirical representation of $\mathrm{p}(h|i,t)$, the posterior probability of the hand configuration given an image and a task. We proceed by decomposing this problem in two. First, we compute regions of the image that enable the task, i.e., regions where the hand can contact the object and yield a grasp configuration that is compatible with the task. In turn, we compute hand configurations (wrist pose/finger preshape) for which the shape of the hand locally matches the shape of the object, and that are within the task regions defined above. The resulting grasps are geometrically consistent, and compatible with the task. Formally, our model involves the three variables i, t and h defined above, and a fourth variable c that is a ternary image mask that represents the task-compatibility of image regions, as *suitable*, *unsuitable*, or *unknown*. A hand configuration h belongs to $\mathcal{H} = SE(3) \times \mathcal{M}$, where $SE(3)$ is the special Euclidean group modeling wrist poses, and $\mathcal{M}$ is a set of discrete hand preshapes such as *power grasp* or *pinch*. A grasp is executed by bringing the hand to a pose and preshape $h \in \mathcal{H}$, then closing the fingers until contact. Both i and t are given. We assume the conditional independence relations represented by the network of Fig. 2 – specifically, that the hand configuration is independent of the task if image regions c are provided. Those assumptions yield a decomposition of the posterior probability of h as

$$\mathrm{p}(h|i,t) = \int \mathrm{p}(h|c,i)\mathrm{p}(c|t,i)\mathrm{d}c, \tag{1}$$

where $\mathrm{p}(c|t,i)$ models contact regions that enable the task, and $\mathrm{p}(h|c,i)$ models the probability that hand configuration h provides a mechanically stable grasps while only contacting visible object areas that belong to c.

Computing task-compatible grasps in a Bayesian fashion as shown in Eq. 1 is scrupulous, but prohibitively expensive. In the following, we instead approximate $\mathrm{p}(h|i,t)$ with

$$\mathrm{p}(h|i,t) \simeq \mathrm{p}(h|c^\star,i), \tag{2}$$

where

$$c^\star = \underset{c}{\operatorname{argmax}}\, \mathrm{p}(c|t,i). \tag{3}$$

Sec. III-B presents the semantic model of Eq. 3. Sec. III-C presents the geometric model of Eq. 2. Finally, Sec. III-D discusses a Markov-chain–Monte-Carlo approach to finding the grasp pose that maximizes Eq. 2.

B. Semantic Model

The estimation of task-compatible contact regions requires high-level knowledge that takes into account both the local geometry as well as the overall structure of a particular object. Deep Convolutional Neural Networks (CNNs) are well-suited to this, as they merge information across a range of receptive field sizes. This property allows them to jointly model both part appearance and location within a larger object. We implemented our task-compatibility model with the MultiNet architecture proposed by Teichmann et al. [34], due to its run-time performance and need for relatively small training sets. MultiNet is a derivative of two of the most popular CNNs: the Visual Geometry Group (VGG) network [29], and the Fully Convolutional Network (FCN) [17].

VGG is useful as it provides a set of weights that were pre-trained on the ImageNet dataset. While the VGG weights were trained on an object classification task, their use has been shown to provide a significant boost over random initialization across a wide variety of tasks. For this work, we use VGG weights as our initial condition for all layers, with the exception of the fully connected layers at the top of the

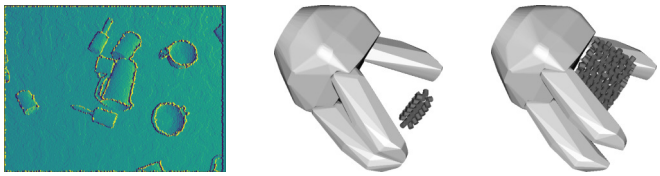


Fig. 3. Left: the x-derivative of a synthetic depth image. Center and right: Two grasp prototypes. The dark gray shapes are point-cloud representations of the prototypes' shapes and their pose relative to the gripper. The left image shows a pinch grasp on a cylindrical object, the rightmost image shows a power grasp on a cubical object.

network and the first convolutional layer at the bottom. The top layers are randomly initialized since our output classes bear no similarity to those of ImageNet, while the bottom layer is randomly initialized since, as explained below, our input images are not in the RGB colorspace. The FCN architecture is important as it introduced the concept of a fully convolutional network; a network which maps directly from input pixels to class-labeled output pixels.

While we use MultiNet as our architecture, we diverge from it on the input side: rather than using RGB images as our input, we use preprocessed depth images. The decision to use geometry, rather than color, as our input space is based on three factors. First, we want to generalize across object instances of the same class, and doing so with color requires a prohibitively large training set. Secondly, while rendering photo-realistic color images is difficult, we are able to render synthetic depth images that are comparatively closer to real depth images. Finally, we wish to generalize similar task-relevant geometries across object classes. For example, if handles are relevant to a task, we want to recognize their importance independent of an object's color, since only the local shape of the handle is important.

The use of depth as an input feature for CNNs is not as well understood as color. While there have been some previous attempts to use it both directly [23], and encoded, as in HHA [13] (which encodes horizontal disparity, height above ground, and angle with gravity), how to take advantage of the rich information depth contains remains an open question. In this work, we select a local gradient encoding which contains three input channels: an approximation of the x-derivative, y-derivative, and gradient magnitude, calculated using Sobel filters. This encoding avoids the normalization and magnitude issues presented by using raw depth, while remaining fast and general (unlike HHA, which requires a floor-plane to be visible). Fig. 3 (left) shows an example of depth x-derivative.

C. Geometric Model

The previous section documented our approach to computing task-compatible contact surfaces. In this section, we discuss the problem of computing hand parameters that enable mechanically-workable grasps. Grasping an unknown object from a single depth image is a difficult problem, largely because we need to place at least one finger on a surface we cannot observe. We address this problem with a part-based approach: we assume the existence of a dictionary of *grasping prototypes*, composed of a prototypical shape, and corresponding grasp parameters. Fig. 3 shows two of the eight prototypes that we are using in our experiments. The grasp parameters considered in this work are composed of the pose of the gripper (with respect to the corresponding shape model), and a hand preshape that is either *pinch grasp* or *power grasp*. To grasp a new object, the robot aligns all prototypes to a point cloud of the scene, and executes the grasp that corresponds to the best-fitting prototype by preshaping the gripper, moving it to the computed pose, and closing the fingers until contact. In effect, fitting 3D prototypes to a partial view implicitly postulates the shape of the object in occluded areas, providing us with a rationale for placing fingers on unseen surfaces. We have shown in previous work how to learn such a dictionary from experience [7]. The next paragraph summarizes the algorithm that allows us to grasp new objects. For further details on this algorithm, we refer the reader to our previous work [7].

We model a prototype's shape with a *surface density* [8]. A surface density is a function $\mathrm{q}(w) : \mathbb{R}^3 \times S^2 \to [0, 1]$ that models a 3D shape probabilistically. Intuitively, if the shape of a prototype is such that the point $\ell \in \mathbb{R}^3$ belongs to its surface, and the local normal is n, then $\mathrm{q}([\ell, n])$ is greater than zero. For a point w that is far from any surface, then $\mathrm{q}(w) = 0$. We encode surface densities nonparametrically with at set of samples, and evaluate them via KDE [8].

This model offers an elegant solution to the prototype alignment problem discussed above. Let us denote the surface density model of the input image i by $\mathrm{q}(w)$, and by $\mathrm{s}_k(w)$ the surface density of prototype k's shape. We define the surface matching score for prototype k at pose x by marginalizing the joint distribution of prototype poses and object surface points, as

$$\mathrm{p}_k(x|i) = \int \mathrm{s}_k(x|w)\mathrm{q}(w)\mathrm{d}w, \tag{4}$$

In this expression, the conditional $\mathrm{s}_k(x|w)$ is defined as

$$\mathrm{s}_k(x|w) = \mathrm{s}_k(w - x), \tag{5}$$

where $w - x$ is the $SE(3)$ transformation of w by x. Intuitively, for a given grasp pose x, $\mathrm{s}_k(x|w)$ is equal to the surface distribution model of prototype i, translated and rotated by x. Eq. 4 measures the overlap between $\mathrm{s}_k(x|w)$ and $\mathrm{q}(w)$, effectively yielding a surface matching score.

The matching score of Eq. 4 ignores the task requirements defined by the semantic model discussed in the previous section. To take those into account, we constraint Eq. 4 to image surfaces that are labeled task-compatible by the semantic model, by replacing $\mathrm{q}(w)$ by a surface density $\mathrm{q}'(w)$ constructed exclusively from points that are labeled positively by the semantic model of Sec. III-B.

D. Joint Maximization of Semantics and Geometry

To plan a grasp, we compute the prototype index $k^\star$ and gripper pose $x^\star$ that maximize Eq. 4, as

$$k^\star, x^\star = \underset{k,x}{\operatorname{argmax}}\, \mathrm{p}_k(x|i). \tag{6}$$

Fig. 4. Top-left: manipulation testbed. Top-right: objects used for training the CNN. Bottom: Test objects.

transport	*affected objects*: All.
	constraints: If object has handle: by the handle. Otherwise, anywhere.
handover	*affected objects*: Objects that have handles.
	constraints: Leave handle available for operator's grip.
pour	*affected objects*: Containers (objects that have an opening).
	constraints: Grasp away from opening.
open	*affected objects*: Containers (objects that have a lid).
	constraints: Grasp away from lid.

TABLE I

FOUR TASKS CONSIDERED IN THIS EXPERIMENT, AND ASSOCIATED CONSTRAINTS.

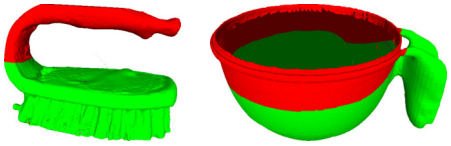


Fig. 5. Two examples of task-constraint labels from the training set. Green and red surfaces are respectively suitable and unsuitable for the task. The left image shows the *handover* constraints for a brush: the robot must leave the handle available for the operator to apply his grip. The rightmost image shows a *pour* annotation: the robot must avoid touching the object near its opening. Similar constraints are defined for all applicable combinations of the four tasks and ten training objects.

The integral of Eq. 4 is not tractable analytically. Instead, we approximate it with Monte Carlo integration [3], [8], as

$$\hat{\mathrm{p}}_k(x|i) \simeq \frac{1}{M} \sum_{\ell=1}^{M} \mathrm{s}_k(x|w_\ell) \quad \text{where} \quad w_\ell \sim \mathrm{q}'(w), \qquad (7)$$

where M is a large numerical constant, and $\mathrm{q}'(w)$ only contains surface points that are labeled positively by the semantic model.

To compute the prototype index $k^\star$ and gripper pose $x^\star$ that maximize Eq. 6, we apply simulated annealing to a Markov chain whose invariant distribution is an increasing power of $\hat{\mathrm{p}}_k(x|i)$. For a discussion of this method, we refer the reader to our previous work [8].

IV. EXPERIMENT

In this section, we explain how we train the CNN of Sec. III-B, we evaluate the performance of the CNN on synthetic data, and we evaluate the applicability of our model to task-oriented grasping on an industrial robot.

A. Task Constraints

To train the CNN of Sec. III-B, one would ideally annotate a set of object images by hand. Unfortunately, CNNs require a large amount of training data, and annotating those data manually is in our case prohibitively costly. Instead, we manually labeled a small set of 3D object meshes, and generated training scenes synthetically.

Fig. 4 (top-right) shows the ten objects that we used to train the network, five of which are from the YCB dataset [4]. In this experiment, we are considering four tasks: transport, handover, pour, and open. Our objective is to plan grasps that enable these tasks. Table I defines the constraints that define our four tasks.

In order to avoid expensive and time-consuming manual human annotations, we chose to use simulated depth images for training. To do this, we encoded task constraints by labeling the vertices of mesh models of the training objects as *suitable* or *unsuitable*, as illustrated in Fig. 5. We then generated arbitrary configurations of the training objects by virtually dropping randomly placed objects onto a plane, using a simulator and a physics engine. We rendered simulated depth images using the BlenSor sensor simulation framework [11], which provided a realistic depth-camera sensor model. This was necessary, as it allowed our synthetic data to emulate traits of the structured-light sensors mounted on the robot, facilitating the direct transfer of trained models to real data. Fig. 6 shows an example of a synthetic training scene and ground-truth labels.

We trained our model on a set of 5000 synthetic training images. For simplicity, we trained a separate network for each of the four tasks listed above (we do not see any obstacle to capturing all tasks in a single network in future work). To avoid overfitting, we interrupted training once the training set loss began to diverge from the validation set loss. The network uses dropout as a regularizer. In the future, we plan to avoid overfitting by continuously

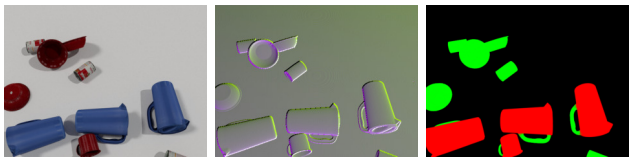


Fig. 6. Left image: the RGB channel of a synthetic training scene, showing objects of Fig. 4. This image is for illustration only; we do not use color in any model discussed in this paper. Center: Depth image derivatives (x and y) and depth gradient intensity, rendered as R, G, and B channels respectively. Right: ground-truth labels for the $transport$ task.

Task	MaxF1	MAP
Transport	0.738	0.789
Handover	0.959	0.986
Pour	0.904	0.959
Open	0.859	0.924

TABLE II

MAXF1 AND MEAN AVERAGE PRECISION (MAP) FOR PIXELWISE TASK-SUITABILITY CLASSIFICATION ON SYNTHETIC VALIDATION DATA.

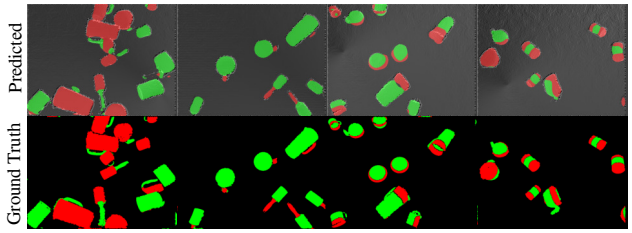


Fig. 7. Task suitability results on synthetic images. From left to right: transport, handover, pour, and open. Examples of classification results demonstrate that the networks are able to learn the different tasks, and can generalize across novel views and configurations of the objects.

generating new viewpoints during training. Training took approximately six hours for each network on a single Titan X GPU, while test evaluation of new images takes 100ms on the same GPU. We then evaluated the ability of the trained models to generalize to new object poses using a separate validation set of synthetic data. Qualitatively, the network learns to recognize the parts of the objects relevant to each task, and labels them appropriately, as can be seen in Fig. 7. Table II provides quantitative results. The networks successfully capture the constraints of all tasks, with better performances for handover, pour and open. This discrepancy is explained by ambiguity in the definition of the *transport* constraints: *grasp on handle if exists, anywhere otherwise*, which leads to contradicting labels for non-handle grasps on objects that have a handle or not. Overall, Fig. 7 and Table II clearly demonstrate the same result: the network is able to accurately label the parts of the objects from novel views and in previously unseen configurations.

B. Task-oriented Grasping

In this section, we evaluate the applicability of our method to task-oriented grasping on an industrial robotic manipulator. The manipulator is composed of a 7-DOF articulated arm, and a three-finger gripper from Robotiq. Depth data are provided by a Kinect 1 camera that is rigidly connected to the robot base. This setup is shown in Fig. 4.

This experiment consisted in executing a set of grasps on novel objects – objects that differed in size and shape from those used for training. Each test proceeded as follows: First, we randomly selected one of the four tasks listed above, and we arbitrarily placed one or several objects on the table. We selected objects such that the task is applicable to at least one object on the table — e.g., if the task is *handover*, at least one object has a handle. We then captured a depth image, zeroed all pixels that corresponded to points located 2cm underneath the tabletop or below, and computed a task-compatibility mask on the resulting depth image using the model of Sec. III-B. Next, we computed the grasp (hand pose and preshape) that maximized the geometric model while complying with the task's constraints (Eq. 6), limiting the pose search space to kinematically-feasible configurations that did not collide with the table or object surfaces recorded by the 3D camera. When multiple objects were present on the table, the maximization step of Eq. 6 effectively decided which object the robot would grasp; the experiment did not require explicit object segmentation. The maximization step took on average 10s on an Intel E5–2687W CPU. Finally, we computed and executed a trajectory that brought the gripper to the desired pose, and closed the gripper until contact. We evaluated the model according to three criteria. The first criterion C1 captured whether the constraints computed by the CNN correctly matched the task's constraints. We evaluated it by inspecting the mask computed by the CNN, using the rules listed in Table I. The second criterion C2 captured the mechanical and semantic success of the grasp. We assessed semantic success by assessing whether the grasp is compatible with the task, and mechanical success by having the robot lift the object off the table. The third criterion C3 represented the robot's ability to transport the object to a basket located 80cm from the center of the workspace.

In a first experiment, we executed 32 tests with a single object on the table. Table III summarizes this experiment, and Fig. 8 shows a set of examples. In Table III, the number of C2 grasps is larger than the number of C1 grasps. This observation is illustrated by Fig. 8d: while the mask predicted for this case correctly rejected the leftmost side of the container's opening, it missed the other side, hence failing C1. The partial failure of the grasp model did however not prevent the grasp from being compliant with the task. In Fig. 8f, the network rejected the handle, which is the only part that the robot is allowed to grasp to execute the *pour* task, whose constraint is to avoid touching the opening of the container. This failure can probably be explained by a lack of sufficiently similar objects in the training set. Fig. 8e shows an example of a C3 failure: the weight of the object overcame the grip during transport.

We repeated the experiment above, with multiple objects on the table instead of only one. Here, C1 captured whether the task constraints of *all relevant objects* were correctly identified by the CNN. We added a criterion C1b that captured whether the task constraints *of the object the robot eventually grasped* was correct. Table IV summarizes this experiment, and Fig. 9 shows a set of examples. The table shows that in approximately half of the tests, at least one object was misclassified. To put this number in perspective, we note that there were on average three objects on the table for each run. The C1b success rate in this experiment is similar to the C1 rate of the single-object case, which demonstrates the robustness of the model to clutter. Fig. 9a shows a case where two objects were misclassified (the two

C1	Valid task-constraint mask	25 / 32	(78%)
C2	Valid grasp (mechanical & semantic)	28 / 32	(88%)
C3	Successful transport	25 / 32	(78%)
C1 ∧ C2		**25 / 32**	**(78%)**
C1 ∧ C2 ∧ C3		22 / 32	(69%)

TABLE III

SUCCESS RATES FOR SINGLE OBJECTS.

C1	Valid task-const. mask (all relevant objects)	12 / 23	(52%)
C1b	Valid task-const. mask (grasped object)	19 / 23	(82%)
C2	Valid grasp (mechanical & semantic)	19 / 23	(82%)
C3	Successful transport	19 / 23	(82%)
C1b ∧ C2		**19 / 23**	**(82%)**
C1b ∧ C2 ∧ C3		15 / 23	(65%)

TABLE IV

SUCCESS RATES FOR THE MULTI-OBJECT EXPERIMENT.

rightmost objects). Fig. 9b is almost a success, but the front half of the object's handle is marked suitable, which is against the constraints of the *handover* task.

While we can observe a distinct performance reduction between synthetic and real data (Fig. 7 and Fig. 8, 9), our results support the networks' ability to gear grasps towards task-compatible regions. The primary factor limiting the accuracy of real-data results is the noise present in the sensor, particularly along object boundaries. Further improvements to the simulation of training data, possibly through the addition of surface material modeling in the BlenSor framework, could help account for this. Finally, unwanted returns occur around the table's edges, because of the absence of those edges in the training set. Future simulations could benefit from dropping objects onto a table, rather than onto a floor, so that the edges of the support surface are visible in training.

V. CONCLUSIONS

We presented a model for task-oriented grasping that jointly exploits a semantic and geometric understanding of the scene. We implemented the semantic model with a set of task-specific CNNs that we trained to identify image regions that the robot is allowed to contact to comply with a given task. We trained the CNN on synthetic scenes randomly generated with hand-labeled mesh models.

The geometric model is a part-based planner that relies on a dictionary of prototypical grasps. Given a task directive, the model searches through scene surfaces that are labeled positively by the task's CNN, to find a prototype pose that maximizes overlap with the scene. The result of this process is a gripper pose and preshape parameters that yield a grasp that is geometrically consistent, and that only contacts object surfaces that will not preclude the task from performing correctly. We trained the semantic model on 5000 scenes, generated with ten hand-labeled objects, and evaluated its applicability both on synthetic data and on an industrial robot. Our results make the generalization capability of our model explicit: despite noisy depth images, we were able to transfer task constraints to objects that differ in size and shape from the training set.

VI. ACKNOWLEDGEMENTS

The authors thank Tristan Thrush for his key contributions to configuring software drivers for the arm, hand and camera.

REFERENCES

[1] L. Antanas, P. Moreno, M. Neumann, R. P. de Figueiredo, K. Kersting, J. Santos-Victor, and L. De Raedt. High-level reasoning and low-level learning for grasping: A probabilistic logic pipeline. *arXiv preprint arXiv:1411.1108*, 2014.

[2] D. Berenson, S. Srinivasa, and J. Kuffner. Task space regions: A framework for pose-constrained manipulation planning. *The International Journal of Robotics Research*, 30(12):1435–1460, 2011.

[3] R. Caflisch. Monte carlo and quasi-monte carlo methods. *Acta Numerica*, 7:1–49, 1998.

[4] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar. The ycb object and model set: Towards common benchmarks for manipulation research. In *International Conference on Advanced Robotics*, 2015.

[5] S. H. Creem and D. R. Proffitt. Grasping objects by their handles: a necessary interaction between cognition and action. *Journal of Experimental Psychology: Human Perception and Performance*, 27(1):218, 2001.

[6] H. Dang and P. K. Allen. Semantic grasping: Planning robotic grasps functionally suitable for an object manipulation task. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

[7] R. Detry, C. H. Ek, M. Madry, and D. Kragic. Learning a dictionary of prototypical grasp-predicting parts from grasping experience. In *IEEE International Conference on Robotics and Automation*, 2013.

[8] R. Detry and J. Piater. Continuous surface-point distributions for 3D object pose estimation and recognition. In *Asian Conference on Computer Vision*, pages 572–585, 2010.

[9] A. H. Fagg and M. A. Arbib. Modeling parietal-premotor interactions in primate control of grasping. *Neural Netw.*, 11(7-8):1277–1303, 1998.

[10] J. J. Gibson. *The Ecological Approach to Visual Perception*. Lawrence Erlbaum Associates, 1979.

[11] M. Gschwandtner, R. Kwitt, A. Uhl, and W. Pree. Blensor: blender sensor simulation toolbox. In *International Symposium on Visual Computing*. Springer, 2011.

[12] S. Gu, E. Holly, T. Lillicrap, and S. Levine. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *IEEE International Conference on Robotics and Automation*, 2016.

[13] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik. Learning rich features from rgb-d images for object detection and segmentation. In *European Conference on Computer Vision*, 2014.

[14] M. Hjelm, C. H. Ek, R. Detry, and D. Kragic. Learning human priors for task-constrained grasping. In *International Conference on Computer Vision Systems*. Springer, 2015.

[15] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal. Dynamical movement primitives: learning attractor models for motor behaviors. *Neural computation*, 25(2):328–373, 2013.

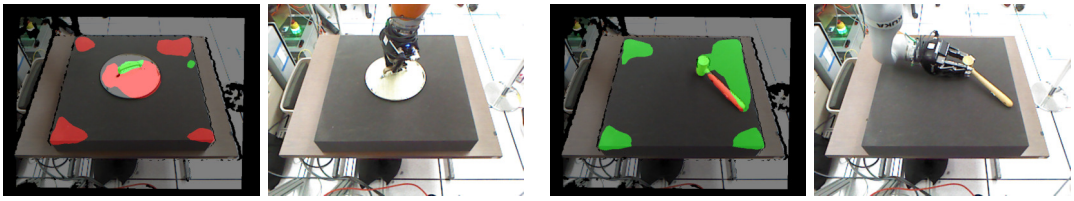
[16] O. Kroemer, E. Ugur, E. Oztop, and J. Peters. A kernel-based approach to direct action perception. In *IEEE International Conference on Robotics and Automation*, 2012.

[17] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

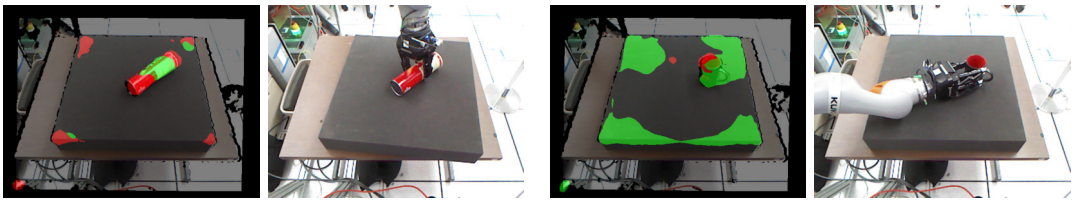
[18] J. Mahler, F. T. Pokorny, B. Hou, M. Roderick, M. Laskey, M. Aubry, K. Kohlhoff, T. Kröger, J. Kuffner, and K. Goldberg. Dex-net 1.0: A cloud-based network of 3d objects for robust grasp planning using a multi-armed bandit model with correlated rewards. In *IEEE International Conference on Robotics and Automation*, 2016.

[19] N. Mavrakis, M. Kopicki, R. Stolkin, A. Leonardis, et al. Task-relevant grasp selection: A joint solution to planning grasps and manipulative motion trajectories. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, 2016.

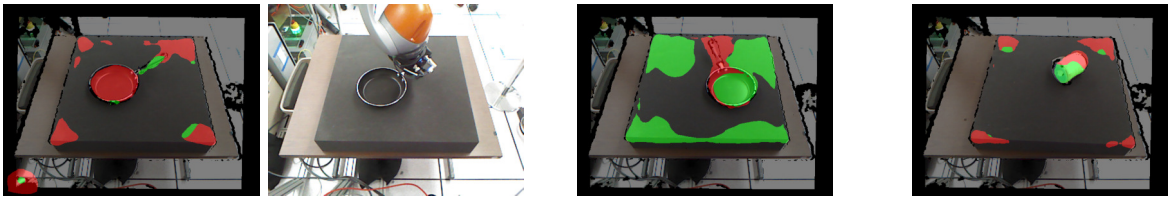
[20] A. D. Milner and M. A. Goodale. Two visual systems re-viewed. *Neuropsychologia*, 46(3):774–785, 2008.



(a) *transport* task, success (b) *handover* task, success

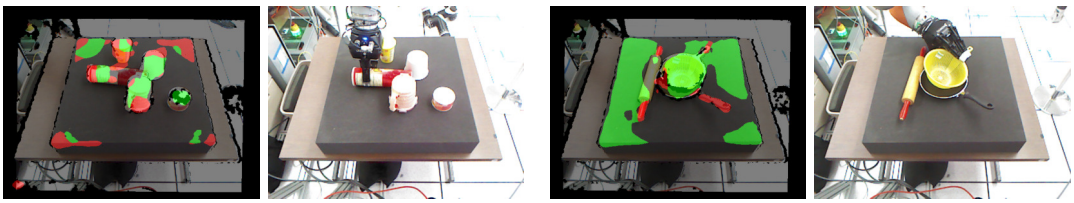


(c) *open* task, success (d) *pour* task, failed C1, passed C2 & C3



(e) *transport* task, passed C1 & C2, failed C3 (f) *pour* task, failed all (g) *pour* task, failed all

Fig. 8. Sample results for the single-object experiment. Labeled images correspond to masks computed by the task-constraint models (green is suitable, red unsuitable).



(a) *open* task, failed C1, passed others (b) *handover* task, failed C1, C1b, C2, passed C3

Fig. 9. Sample results for the multi-object experiment. Labeled images correspond to masks computed by the task-constraint models (green is suitable, red unsuitable).

[21] A. Myers, C. L. Teo, C. Fermüller, and Y. Aloimonos. Affordance detection of tool parts from geometric features. In *IEEE International Conference on Robotics and Automation*, 2015.

[22] A. Nguyen, D. Kanoulas, D. G. Caldwell, and N. G. Tsagarakis. Detecting object affordances with convolutional neural networks. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016.

[23] J. Papon and M. Schoeler. Semantic pose using deep networks trained on synthetic rgb-d. In *IEEE International Conference on Computer Vision*, 2015.

[24] G. Rizzolatti and G. Luppino. The cortical motor system. *Neuron*, 31(6):889–901, 2001.

[25] A. Roy and S. Todorovic. A multi-scale cnn for affordance segmentation in rgb images. In *European Conference on Computer Vision*, 2016.

[26] E. Sahin, M. Cakmak, M. R. Dogar, E. Ugur, and G. Ucoluk. To afford or not to afford: A new formalization of affordances towards affordance-based robot control. *Adaptive Behavior*, 2007.

[27] A. Saxena, J. Driemeyer, and A. Y. Ng. Robotic Grasping of Novel Objects using Vision. *International Journal of Robotics Research*, 27(2):157, 2008.

[28] A. Saxena, L. Wong, and A. Ng. Learning grasp strategies with partial shape information. *Association for the Advancement of Artificial Intelligence*, 2008.

[29] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[30] D. Song, C. H. Ek, K. Huebner, and D. Kragic. Multivariate discretization for bayesian network structure learning in robot grasping. In *IEEE International Conference on Robotics and Automation*, 2011.

[31] H. O. Song, M. Fritz, C. Gu, and T. Darrell. Visual grasp affordances from appearance-based cues. In *IEEE Workshop on Challenges and Opportunities in Robot Perception*, 2011.

[32] A. Srikantha and J. Gall. Weakly supervised learning of affordances. *arXiv preprint arXiv:1605.02964*, 2016.

[33] A. Stoytchev. Behavior-grounded representation of tool affordances. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005.

[34] M. Teichmann, M. Weber, M. Zoellner, R. Cipolla, and R. Urtasun. Multinet: Real-time joint semantic reasoning for autonomous driving. *arXiv preprint arXiv:1612.07695*, 2016.

[35] N. Vahrenkamp, L. Westkamp, N. Yamanobe, E. E. Aksoy, and T. Asfour. Part-based grasp planning for familiar objects. In *IEEE/RAS International Conference on Humanoid Robots*, 2016.

[36] K. M. Varadarajan and M. Vincze. Afrob: The affordance network ontology for robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.