

Representations for Cross-task, Cross-object Grasp Transfer

Martin Hjelm

Renaud Detry

Carl Henrik Ek

Danica Kragic

Abstract—We address the problem of transferring grasp knowledge across objects and tasks. This means dealing with two important issues: 1) the induction of possible transfers, i.e., whether a given object affords a given task, and 2) the planning of a grasp that will allow the robot to fulfill the task. The induction of object affordances is approached by abstracting the sensory input of an object as a set of attributes that the agent can reason about through similarity and proximity. For grasp execution, we combine a part-based grasp planner with a model of task constraints. The task constraint model indicates areas of the object that the robot can grasp to execute the task. Within these areas, the part-based planner finds a hand placement that is compatible with the object shape. The key contribution is the ability to transfer task parameters across objects while the part-based grasp planner allows for transferring grasp information across tasks. As a result, the robot is able to synthesize plans for previously unobserved task/object combinations. We illustrate our approach with experiments conducted on a real robot.

I. INTRODUCTION

Interaction with objects through grasping and manipulation is an important ability of robots operating in natural settings. A robot may be equipped with some initial knowledge of how to grasp and manipulate certain types of objects, but the ability to generalize across objects and tasks will provide it with a much wider flexibility in interaction with the environment.

In this paper, we focus on generalization principles that conceptualize the modeling of relations between objects, grasps and tasks. We use this as a basis for knowledge transfer in situations where either the task and/or object may not be known a-priori.

An example of the generalization principles we want to achieve is: *"Having seen a spoon used for stirring, now given the choice of a fork or a bottle which one would be best to perform the same action with and how would we grasp it to perform the task?"* We see this as a representation-learning problem. This means that we want to learn transforms of the sensory input that enables the agent to reason about, explanatory factors for whether an object affords a task or not, and the connection these factors have with the strategies for executing the task. This closely relates to the idea of encoding object affordances [1].

M. Hjelm, C. H. Ek and D. Kragic are with the Centre for Autonomous Systems and the Computer Vision and Active Perception Lab, CSC, KTH Royal Institute of Technology, Stockholm, Sweden. Email: {martinhjelm, chek, danik}@csc.kth.se. R. Detry is with the University of Liège, Belgium. Email: renaud.detry@ulg.ac.be.

This work was supported by the Swedish Foundation for Strategic Research, the Belgian National Fund for Scientific Research (FNRS), the Swedish Research Council, and the EU project TOMSY (IST-FP7-Collaborative Project-270436).

We formulate the transforms through semantic descriptors of the object, that is, what humans simply would call attributes. Attributes encode concrete or abstract concepts of objects such as a handle, opening, roundness, elongatedness, etc.

We want the robot to incrementally discover the attribute requirements, the explanatory factors, which are necessary for a task while using a minimum of training data. To that end, we use an inductive logic along the lines of: *"The objects I have seen that affords throwing were all round and therefore to throw an object it must be round."* Practically this means, giving the robot the ability to classify different degrees of roundness as well as demonstrating negative examples of non-round objects.

Once the robot has induced that an object affords a task, it must transfer its obtained experience from executing the grasp to execute the grasp on the new object. Our solution is to modularize the task grasp model into constraints relating to the task and constraints relating to the gripper and object.

To learn the task constraint model we employ a semi-supervised approach using observations of the task being performed on objects. The task constraints relate the observed grasps to the attributes of the object, and forms beliefs about good grasping regions for the task. The gripper constraints encode local shape descriptors that generalize between objects and allows for the transfer of grasps between objects.

To execute a task-specific grasp the planner searches in the grasping regions specified by the task constraints. It uses the part-based planner to find a hand placement in the regions that is compatible with the local object shape; that will produce a stable grasp.

II. CONTRIBUTIONS AND RELATED WORK

A recently identified problem of importance is that of transferring object and task-specific grasp knowledge. There are works that approach the problem using synthetic data but none, which to the best of our knowledge, treats it fully in a real world setting.

Our approach to achieving an understanding of the prerequisites for a possible transfer has been to employ a simple logic that requires as few training examples as possible. Similar to our work, Abdo et al. [2] developed a strategy for training the robot to recognize preconditions and effects for actions based on small set of training data. Chao et al. [3] used learning from demonstration (LfD) to learn grounded concepts based on low-level sensory experience. The authors use these concepts to develop prerequisites and expectations of changes that are needed to complete given tasks. Krüger

et al. [4] formulated the object action complexes (OAC) framework as a basis for representing and reasoning about sensory-motor experience.

Other recent work concentrates specifically on developing graphical models for encoding knowledge, and perform inference, under uncertainty. Song et al. [5] trained a Bayesian Network (BN) that relates object, task, and constraint features for a set of objects and tasks. To find a suitable grasp the robot queries the BN conditioned on the task and object features. Contrary to our method, this approach requires a substantial amount of synthetic training data, 1800 data points for three tasks, and cannot generalize beyond learned categories, requiring complete supervision.

Madry et al. [6] enhanced the BN in [5] by an object categorization system that helps refine the grasp queries. However, the tradeoff is instead a dependency on a constructed categorization structure. In the current paper, we consciously shift away from higher domain knowledge relying on a simple inductive system paired with LfD, allowing us to use a small set of training data.

Dang et al. [7] introduced a data driven approach to modeling the constraints inherent in a task. The authors build a representation of an object using depth images and manually specify task-specific parameters on it. The parameters form a task constraint map over the representation that is queried when grasping the object for a task. Our method is similar in that we decompose the grasp problem into task constraints and stability constraints. However, instead of requiring a complete representation of the object, we decompose it into parts relevant for the task and leave the agent to use its acquired knowledge to reason about which parts are relevant for completing the task. This gives the agent the possibility to perform tasks on objects beyond the already known.

Other authors have approached the topic of task constraints through decomposition by parts, the physics of objects, and analysis of synthetic datasets. Ying et al. [8] used task constraints to prune possible grasp candidates on an object. They formulate the task constraints such that the grasp must support the object against gravity in vertical and near-vertical orientations.

Sahbani et al. [9], motivated by [10], decomposed objects into parts. Similar to our work, the authors make a set of assumptions about the design of everyday objects and the grasping strategy for them. From a synthetic dataset, they extract *graspable parts* from their parts representation and train a classifier to recognize them. Once the classifier finds a graspable part on an object, they use an analytical approach for finding a force-closure grasp on it.

Aleotti et al. [11] also used decomposition, and LfD to build a task-oriented planning system. The authors use a topological representation to do decomposition and recognition of objects. The robot learns grasps by LfD on a synthetic dataset. They construct task specific grasp maps for the objects and use the learned grasps as matching criteria. To grasp the object for a task they sample grasps from the grasp map focusing on the task-specific parts of the object. Though good at finding task-specific grasps, the acquired

generalization capability is nonetheless limited to objects with similar shape.

Although quite successful in finding good grasps, it is unclear how [7], [9], and [11] translate into a real environment with no full 3D representations and only partial views of the object. In addition, the analytical approach to finding force-closure grasps with no additional learning could prove difficult in a real setting.

III. MOTIVATION AND SYSTEM OUTLINE

The concrete questions that motivate us are: How can we reason about object properties and tasks in a simple way while still giving the robot an adequate ability to induce possible transfers? How can we use the acquired skill set and experience from performing the task as a basis to perform a suitable task-specific grasp on a novel object? This is what we will refer to as a transfer: the adaption of the learned task-specific grasp parameters applied to novel objects.

A. Attribute Representation

We take a geometrical approach to encoding the attributes and model the reasoning aspect based on measures of proximity. Attributes are modeled as feature extraction functions, A_k , acting on the sensory input of an object, O_i , where k and i is the attribute and object index. T represents tasks; here meaning task-specific grasps.

To enforce the proximity measure we formulate the attribute functions, A_k , such that attributes that are similar are close in the feature space. Formally, we write,

$$A_k : O \mapsto X_k, \quad X_k \in \mathbb{D}_k, \quad (1)$$

where \mathbb{D}_k is the attribute space, which can be either discrete or continuous. We refer to section IV for the specifics related to the implementation.

B. Induction

Following the observation that affordances relate to attributes, we make the assumption that we can explain all affordances of an object by reasoning about its attributes. Obviously, this is not always true but recurrent enough and serves our purpose of using minimal domain knowledge. To this end, we use simple inductive logic to reason about affordances through attribute presence and similarity.

For example, if we want to teach a robot that objects that affords throwing require an attribute such as roundness, we demonstrate the throwing task on a ball as a positive example and a block as a negative example. To induce if a novel object affords throwing, the robot considers the proximity to the positive and negative examples it has seen. In this paper, we use the simple nearest neighbor Euclidian distance to infer whether the task is transferrable or not. We illustrate this in Fig. 4 in Section V.

Finally, to falsify the notion of roundness as a requisite for throwing, the teacher needs to show the robot positive examples of non-round objects that affords throwing. If the distance is lower than some minimum threshold, ϵ , the

attribute is considered irrelevant for transfer. We summarize the logic in the following algorithm/procedure:

TRANSFER BY ATTRIBUTE()

- 1 **Learning:**
- 2 Observe $\mathbf{A}_1 = [A_1, A_2, \dots]$ on O_l during task T
- 3 Observe $\mathbf{A}_e = [A_1, A_2, \dots]$ on O_e during task $\neg T$
- 4 **Induction:** Is T transferrable to O_s ?
- 5 Observe $\mathbf{A}_s = [A_1, A_2, \dots]$ on O_s .
- 6 If $\forall i D(\mathbf{A}_{1i}, \mathbf{A}_{si}) \leq D(\mathbf{A}_{ei}, \mathbf{A}_{si})$
- 7 then O_s affords T

C. Generalizing Grasp Parameters

To generalize the grasp parameters we modularize the task grasp model into *task constraints* – the constraints inherent in the task – and *gripper constraints* – the constraints inherent in the shape of the object and the shape of the gripper. The exact specifications of the task and gripper constraints are dependent on the task, attributes, experience, and gripper.

Under the assumption that attributes determine the affordances of an object, we model the constraints in relation to each attribute. For example, if we want to teach the robot not to cover the opening when grasping a bottle to pour from, we encode openings as an attribute. We then record previous grasping points and their position to the opening such that they are object-scale invariant. The robot thus has no higher-level understanding of what an opening is and how it works. In fact, we show in Section V that under simplified conditions we can rely on another much simpler attribute to learn that we should grasp away from the opening when grasping for pouring.

We model the gripper constraints based on our previous work presented in [12]. The central observation of [12] is that objects often share local shape and by generalizing them, we can grasp similar shapes on other objects. For example, segments of cylindrical shape are generalizable to one prototypical shape, while cuboid shaped segments are generalizable to another prototypical shape, and so on. If we know how to grasp these different kinds of prototypes then we should be able to grasp shape segments similar to them.

To envision how these modules work together, a simple example will suffice: the robot wants to grasp a milk carton to pour milk into a glass. It has previously learned how to grasp a cuboid shaped milk carton to move it from the fridge and put it on the table. It has also learned how to pour a glass of juice from a bottle by grasping it on the upper part. Here the task constraints are: grasping on the upper part of the object to pour, and the gripper constraints are: how to grasp cuboid shapes. To find a stable grasp for pouring the milk into the glass the robot combines these two constraints. It searches for a good grasp on the upper part of the carton combining it with the experience gained from grasping the cuboid shape segment. We visualize the example in Fig. 1

In summary, task constraints enforce the search for a suitable grasp that matches previously observed constraints for the task while the gripper constraint seeks for a suitable grasp that matches the local shapes we have grasped before.

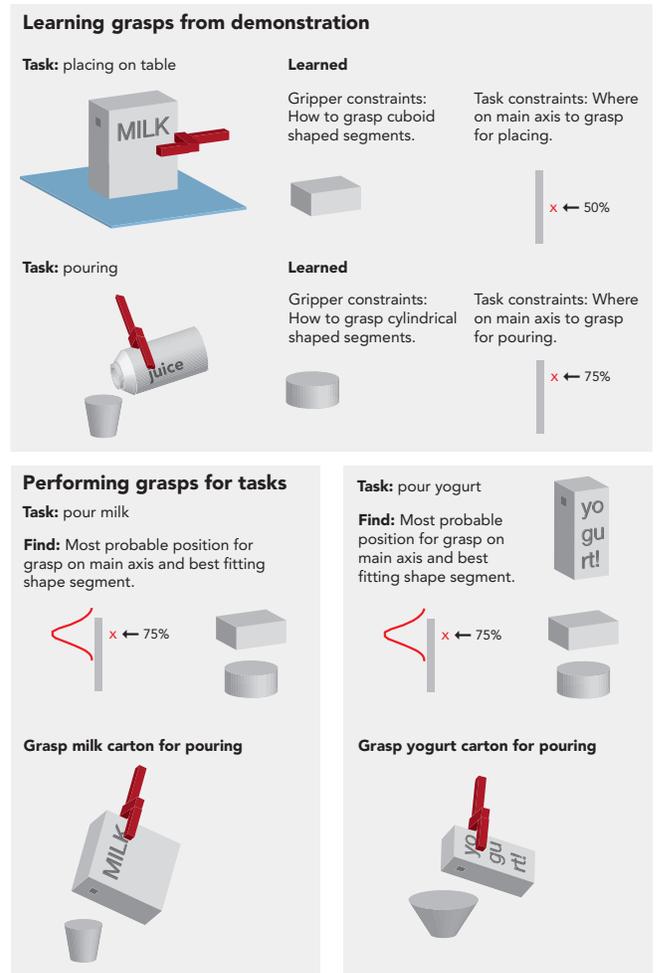


Fig. 1: Simplified outline of our grasp-planning module. First, the robot learns to grasp specific shape segments (gripper constraints) and good task-specific grasping positions (task constraints) on the main axis using the elongation attribute. To perform the task-specific grasp on an object other than the one used in the learning phase, the robot matches learned shape segments with the belief of good grasping regions for the task.

IV. METHODOLOGY

The key to making the induction and transfer as flexible and adaptive as possible lays in formulating attributes and constraints as object-invariant as possible but without compromising the quality and ability to repeat the learned tasks. In the following subsections, we specify an attribute and build on that to create a link to the task constraints.

A. Attributes

In this paper, we use the elongatedness of objects, an important and common attribute of most objects, especially containers and tools. Elongation is interesting since on most elongated objects there are many possibilities to finding suitable grasps even with a two-fingered gripper. We define the elongatedness attribute function as,

$$A_{Elongated} : O \mapsto \left(\frac{b}{a}, \frac{c}{a}\right), \quad (2)$$

where a is the major axis length of the object and, b and c are the minor axes lengths.

B. Modeling Task Constraints

We encode task constraints by modeling the viable positions of a grasp along the main axis of the object. The main axis of an object is computed from the principal components of the (possibly partial) point cloud of the object. To generalize and enforce object invariance, the axis is normalized to unit length. We model the constraints imposed by a task with a probability density function $t(y)$ defined on the segment $[0, 1]$. The function t yields a high value at fractions y of an object’s main axis that correspond to a viable grasp.

Task constraints are modeled with a Gaussian Mixture Model (GMM), as

$$t(y) = \sum_{n=1}^N \pi_n \mathcal{N}(y|\mu_i, \sigma_i), \quad (3)$$

where μ_i ’s and σ_i ’s are the model’s parameters.

Given an object’s (partial) point cloud O , the compatibility of a gripper pose $x \in SE(3)$ with a task is written as

$$f_1(x|\theta, O) = \sum_{n=1}^N \pi_n \mathcal{N}(g(x, O)|\mu_i, \sigma_i), \quad (4)$$

where θ is the concatenation of the parameters μ_i and σ_i characterizing this particular task, and g is a function that takes the object representation and the gripper pose x and returns the projection of x on the object’s main axis (after normalization to unit length).

Below, in our experiments, the number of components, N is equal to the number of grasps observed for a particular task. Each component has a mean equal to the fraction of an object’s main axis at which a grasp is demonstrated. We set the variance σ_i manually so that 95% of the probability mass is within $\pm 10\%$ of the mean. We formulate the component probabilities, π_n , as follows,

$$\pi_n = \frac{l_n}{l_g}, \quad (5)$$

where l_n is the length of the main axis of the n -th object, and l_g is the fraction of that axis at which a grasp is demonstrated. Our motivation for this is that when the object is taller than the palm of the gripper it will be more precise than when the gripper is of about the same height as the object; and should therefore be more important. We normalize the component probabilities to sum to one.

C. Modeling Gripper Constraints

Gripper constraints are modeled with grasping prototypes [12] (see Section III-C). We teach a set of grasps to the robot, and we let the robot learn the shape of parts by which objects are recurrently grasped. The result is a dictionary of prototypes that describe object parts, and how to grasp each of these parts. Prototypes allow us to measure whether a given gripper pose x fits the object, by measuring the similarity between each of the prototypes and the shape of the object near the grasping site x . If the shape of one

prototype resembles the shape of the object at x , it is likely that x is a viable gripper pose.

We model the shape of object parts with *surface densities* [13], which correspond to probability density functions defined over surface points and surface normals. Surface densities are thus defined over $\mathbb{R}^3 \times S^2$. We estimate the value of a surface density at a specific point via kernel density estimation, using an isotropic Gaussian kernel for position and two antipodal von-Mises Fisher distributions [14] for orientation.

Our probabilistic setting allows us to formulate an elegant solution to the surface similarity problem discussed above. Let us denote the surface density model of an object O by q . The shape similarity between an object and the i^{th} prototype in pose x is obtained by marginalizing the joint distribution of prototype poses (i.e., gripper poses) and object surface points, as

$$f_2(x|i, O) = \int p(x|i, w)q(w)dw. \quad (6)$$

In the equation above, the conditional pose probability $p(x|i, w)$ is simply given by

$$p(x|i, w) = q(w - x), \quad (7)$$

where $w - x$ corresponds to the rigid transformation of w by x . Intuitively, for a given grasp pose x , $p(x|i, w)$ is equal to the surface distribution model of prototype i , translated and rotated by x . In other words, Eq. 6 is the value at x of the $SE(3)$ cross-correlation of the surface density of part i and $q(w)$. Eq. 6 gives us measure of the overlap between $p(x|w)$ and $q(w)$.

D. Performing the grasp

To find the most suitable prototype and task-specific grasp we simultaneously maximize the product of the task constraints and the gripper constraints

$$x, i = \underset{x, i}{\operatorname{argmax}} f_{1+2}(x|i, \theta, O), \quad (8)$$

with

$$\begin{aligned} f_{1+2}(x|i, \theta, O) &= f_1(x|\theta, O)f_2(x|i, O) \\ &= \int f_1(x|\theta, O)p(x|i, w)q(w)dw \end{aligned} \quad (9) \quad (10)$$

This integral is intractable yet equivalent to the expectation of $f_1 p$ under q , i.e., $E_q[f_1 p_i]$. We therefore approximate it by sampling from q ,

$$f_{1+2} \simeq \frac{1}{M} \sum_{k=1}^M f_1(x_k|\theta, O)p(x_k|i, w), \quad x_k \sim q. \quad (11)$$

We perform the sampling using Monte Carlo methods in accordance with [13].

V. EXPERIMENTS

We have designed the experiments to provide a proof of concept that illustrates the main contributions of our approach: the use of minimal data, learning of task constraint



Fig. 2: Objects used in the experiment.

	$\left(\frac{b}{a}, \frac{c}{a}\right)$	$\frac{l_o}{l_g}$
Mashed potato cuboid	(0.76, 0.28)	2.76
Cleaning spray cylinder	(0.34, 0.19)	2.75
Salt cylinder	(0.46, 0.16)	2.02
Salt cuboid	(0.54, 0.26)	1.93

Fig. 3: Left column, the elongatedness attribute value of each object, that is, the relative size of the main axis to the minor axes. Right column, the relative size between the object’s main axis height and the gripper palm height.

parameters, the generalization of these parameters, the induction of possible transfers, and the realization of the actual grasps.

For our experiments, we decide on three tasks: pouring, storing, and shaking; all being tasks that are closely related to the main axis attribute. For each of the tasks we choose one optimal global grasp position on the main axis. The objects we use are: a box of mashed potatoes, cleaning spray, a cuboid shaped saltshaker and a cylindrical shaped saltshaker (Fig. 2).

In the experiments, the objects are placed standing up on a flat surface. We record a point cloud of the scene using a Kinect and subsequently segment out the object. We find the main axes using PCA on the segmented object.

A. Learning from Demonstration

To learn the grasp for pouring we let the robot grasp on the upper part of the main axis; close to the pouring direction.

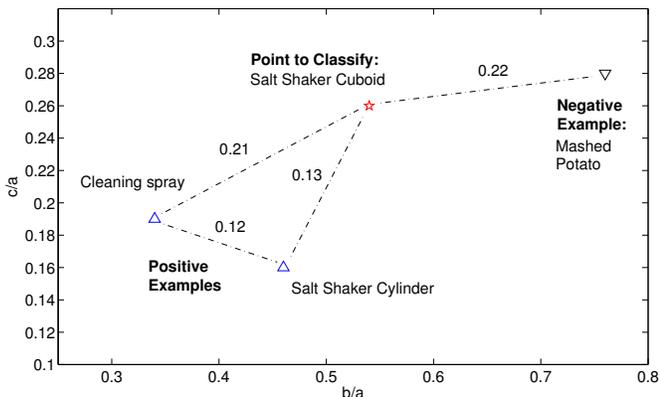


Fig. 4: Determining a possible transfer for pouring. Having observed two possible examples and one negative example for pouring, the robot tries to induce if an object fits the prerequisites for transfer. Using the nearest neighbor distance, the agent induces it as a possible transfer and attempts the grasp using the learned gripper and task constraints for pouring. All distances in the plot are the point wise Euclidian distances and the xy-axes specifies the ratios of the minor axes to the major axis of the objects.

	Store	Pour	Shake
Mashed potato cuboid	21%	80%	51%
Cleaning spray cylinder	20%	81%	44%
Salt cylinder	29%	66%	46%
Salt cuboid	28%	69%	45%

Fig. 5: Matrix of learned grasping points as the relative position on the main axis to perform a specific task. The task constraints are reflected in the values. For storing, we have a low value since we want to store on a high shelf; for pouring, we have a high value meaning we want to grasp close to the opening; and for shaking, we grasp at the mid point of the main axis close to the center of mass.

	Store	Pour	Shake
Mashed potato cuboid	Obs. 21%	Obs. X	Per. 44%
Cleaning spray cylinder	Per. 31%	Obs. 64%	Per. 50%
Salt cylinder	Per. 58%	Per. 50%	Obs. 46%

Fig. 6: Matrix of learned and performed relative grasping points on the main axis for a specific task. Obs. denotes a value observed under the learning phase, Per. specifies that the value is inferred, and X denotes a negative example.

This makes sense since tilting the container for pouring will make the center of mass move forward to the top of the bottle.

When storing objects the grasping point depends on the position of the robot’s arm with respect to the height of the storing place. In this experiment, we want to store on a shelf high up and to comfortably reach the shelf we must grasp on the lower part of the object.

To shake an object, we assume evenly distributed mass, and grasp at the center on the main axis since we want a stable grasp at the center of mass.

To learn the relative grasping points for each task we place the object in the gripper, in the positions we specified. We record the relative grasp position on the main axis in the object’s coordinate system and add the values to the task constraint function.

In this paper, we are mainly concerned with the induction and the task constraints. Therefore, we defer learning of the shape segment prototypes for the gripper constraints and make use of the prototypes already learned in [12].

B. Inducing transfer

In our first experiment, we want the robot to induce that an object is pourable only if it is elongated. We first let the robot observe the grasps for pouring from the spray and the salt cuboid. After this, we show it the mashed potato cuboid as a non-pourable object. The robot records the elongation attribute for both the positive and negative examples.

We put the salt cylinder in front of the robot and order it to grasp for pouring. The robot finds the object’s main axes, and using the nearest neighbor distance, induces that the salt cylinder affords pouring, and is ample for transfer. We illustrate the decision in Fig. 4 and point out that just one of the positive examples is enough to perform the induction.

The mashed potato cuboid obviously affords pouring as seen in Fig. 5 and Fig. 7. By labeling it as a positive



Fig. 7: Illustration of the transfer. Green images show observed task-specific grasps, while red shows inferred task-specific grasps. The columns show the tasks in the following order: store, pour and shake.

example we can reduce or disprove the robot’s belief that only elongated objects affords pouring.

C. Transferring grasps

Having internalized the task constraint values from the demonstration, and induced if an object affords the task, we now want it to perform the task specific grasp on the object.

To perform the grasp the robot runs the task and gripper constraint maximization on the segmented object and once it finds a good candidate grasp position, it executes the grasp. The learned and executed grasps relative grasping points on the main axis can be found in Fig. 6.

If we analyze the inferred grasping points in Fig. 6 we see that by using the task constraints we have learned the robot to grasp away from the opening when grasping for pouring; assuming the opening is at the top. The same applies to storing and shaking. Thus, we have under much simplified conditions, by using a simple task constraint formulation, managed to achieve transfers one would assume requires much more complex constraints.

Further analyzing the data points, we see the limitations due to relative object and gripper sizes. We can also see the physical limitations of the robotic arm and gripper in the environment also having an impact on the results as they need to move so as to not crash into the table; the grasp position is therefore less close to the learned one.

VI. CONCLUSION & FUTURE WORK

We have presented a framework that facilitates transfer of task-oriented grasps across different objects. The key aspect of our approach is its ability to work with a limited amount of training data, by focusing on generalization properties. The proposed method casts generalization as a representation problem so that we can use geometrical notions such as proximity as a proxy for reasoning about similarity and transferability.

In this paper, we do not learn the representations from data but we define them by hand. Clearly, this will not be feasible beyond the simple tasks and objects used in our experiments. Our future work revolves around learning representations directly from data with minimal supervision, where we will build on our previous work on learning clustered representations from continuous data in a generative manner [15]. Metric learning for task of domain adaptation [16] also shows interesting avenues.

To scale up the experiments to more complex and varying shapes, we need to accommodate a wider variety of task constraints. These constraints naturally need to be as object invariant as possible and take into consideration circumstances such as the embodiment of the agent, limitations of sensors, etc.; that are not fully accounted for in the current form. Formulating constraints is not difficult as there is a multitude of cues humans use: color, form, etc., that can be drawn upon. The core of the problem instead lies in the abstractions one chooses, how robust and discriminative they are for the configuration at hand.

REFERENCES

- [1] J. J. Gibson, “The Ecological Approach to the Visual Perception of Pictures,” *Leonardo*, vol. 11, pp. 227–235, July 1978.
- [2] N. Abdo, H. Kretzschmar, L. Spinello, and C. Stachniss, “Learning manipulation actions from a few demonstrations,” in *ICRA*, pp. 1268–1275, 2013.
- [3] C. Chao, M. Cakmak, and A. L. Thomaz, “Towards grounding concepts for transfer in goal learning from demonstration,” in *ICDL*, pp. 1–6, 2011.
- [4] N. Krüger, C. Geib, J. Piater, R. Petrick, M. Steedman, F. Wörgötter, A. Ude, T. Asfour, D. Kraft, D. Omrčen, A. Agostini, and R. Dillmann, “Object–Action Complexes: Grounded abstractions of sensory–motor processes,” *Robotics and Autonomous Systems*, vol. 59, no. 10, pp. 740–757, 2011.
- [5] D. Song, C. H. Ek, K. Huebner, and D. Kragic, “Multivariate discretization for Bayesian Network structure learning in robot grasping,” in *ICRA*, pp. 1944–1950, 2011.
- [6] N. Madry, D. Song, and D. Kragic, “From object categories to grasp transfer using probabilistic reasoning,” in *ICRA*, pp. 1716–1723, 2012.
- [7] H. Dang and P. K. Allen, “Semantic grasping: Planning robotic grasps functionally suitable for an object manipulation task,” in *IROS*, pp. 1311–1317, 2012.
- [8] L. Ying, J. L. Fu, and N. S. Pollard, “Data-Driven Grasp Synthesis Using Shape Matching and Task-Based Pruning,” *TVCG*, vol. 13, no. 4, pp. 732–747, 2007.
- [9] A. Sahbani and S. El-Khoury, “A hybrid approach for grasping 3D objects,” in *IROS*, pp. 1272–1277, 2009.
- [10] I. Biederman, “Recognition-by-components: A theory of human image understanding,” *Psychological review*, vol. 94, no. 2, pp. 115–117, 1987.
- [11] J. Aleotti and S. Caselli, “Part-based robot grasp planning from human demonstration,” in *ICRA*, pp. 4554–4560, 2011.
- [12] R. Detry, C. H. Ek, M. Madry, and D. Kragic, “Learning a dictionary of prototypical grasp-predicting parts from grasping experience,” in *ICRA*, pp. 601–608, 2013.
- [13] R. Detry and J. Piater, “Continuous Surface-point Distributions for 3D Object Pose Estimation and Recognition,” in *ACCV*, pp. 572–585, 2010.
- [14] R. Fisher, “Dispersion on a Sphere,” *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 1953.
- [15] M. Hjelm, C. H. Ek, R. Detry, H. Kjellström, and D. Kragic, “Sparse summarization of robotic grasping data,” in *ICRA*, pp. 1082–1087, 2013.
- [16] B. Geng, D. Tao, and C. Xu, “DAML: Domain Adaptation Metric Learning,” *TIP*, vol. 20, no. 10, pp. 2980–2989, 2011.