On Transferability of grasp-affordances in data-driven grasping

Jørgensen, Jimmy Alison; Ellekilde, Lars-Peter; Kraft, Dirk; Petersen, Henrik Gordon; Krüger, Norbert

Published in:
Proceedings of the RAAD 2013

Publication date:
2013

Document version
Early version, also known as pre-print

Citation for published version (APA):

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 24. Jul. 2018
On Transferability of grasp-affordances in data-driven grasping

Jimmy A. Jørgensen, Lars-Peter Ellekilde, Dirk Kraft, Henrik G. Petersen, Norbert Krüger

The Maersk Mc-Kinney Moller Institute, Faculty of Engineering
University of Southern Denmark, 5230 Odense, Denmark
E-mail: {jimali, lpe, kraft, hgp, norbert}@mmmi.sdu.dk

Abstract. It has become a common practice to use simulation to generate large databases of good grasps for grasp planning in robotics research. However, the existence of a generic simulation context that enables generation of high quality grasps that can be used in several different contexts such as bin-picking or picking objects from a table, has to our knowledge not been discussed in the literature.

In this paper we investigate how well the quality of grasps simulated in a commonly used “generic” context transfer to a specific context where the object is placed on a table. We generate a large database of grasp hypothesis for several objects, which we then evaluate in different dynamic simulation contexts eg. free float (no gravity, no obstacles), standing on table and lying on table. We present a comparison on the intersection of the grasp outcome space across the different contexts and quantitatively show that to generate reliable grasp databases, it is often required to use context specific simulation.

Keywords. Robotic Grasping, Grasp-affordances, Dynamic Simulation, Data-driven grasping

1. Introduction

For more than a decade, data-driven grasp planning approaches have been used in the research community and the main focus has been on how to online select good grasps from a grasp-database generated using a heuristic or simulation (Morales et al., 2006; Berenson et al., 2007; Goldfeder et al., 2009a,b; Goldfeder and Allen, 2011). Only little attention has been put into the simulated context in which these databases were generated and how well the generated grasps actually perform in contexts, which are different from the simulated context, e.g. grasps have been generated in a free floating environment and then applied when an object is placed on a table.

Furthermore, object pose uncertainties due to model and sensor limitations may negatively influence the grasp execution. This can be compensated for in the offline calculation of the grasps (Kim et al., 2012; Weisz and Allen, 2012; Krüger et al., 2012) by evaluating either the quality of neighboring grasps or the quality of the neighborhood grasp contacts, where a neighboring grasp can be simulated by applying a small displacement to the object before executing the target grasp in simulation. The basic idea is that a high average quality of grasps in the neighborhood of the target grasp reflects a high robustness toward uncertainties in the execution of the target grasp.

The above approach of calculating the robustness of a grasp target by evaluating the outcome of the neighborhood using small perturbations, assumes that the calculated neighborhood of grasps - whether done in simulation or with another heuristic - captures the same neighborhood as if the grasps were executed in the real world. This is problematic since the success of a grasp executed in the real world is highly dependent on the environmental context, e.g., when the object is standing on a table, lying in water or leaning against a wall. The obvious possible collisions between gripper and environment does not pose an issue since these can be effectively handled.

However, the interaction forces between the object to be grasped and the environment may differ substantially in different contexts, which may result in different outcomes of attempting the same grasp. The reason is that pose uncertainties imply that the fingers will be unable to synchronously place the fingers at the target contacts on the object, which will cause the object to move during the grasp. The movement will be constrained by the environment and hence different environments/contexts may influence the success of a specific grasp attempt.

In this paper we investigate the transferability of grasp successes from one simulated context to another. Our motivation is not only to understand how the simulated context effects the success of a given grasp,
but also to understand how the success neighborhood of a grasp is affected when transferred to a new context. The influence on the neighborhood of a grasp is important for several reasons:

1. **Grasp quality estimation**: the neighborhood is used to calculate the robustness quality of a grasp, which describe how much pose uncertainties effect the success likelihood of a grasp.

2. **Continuous grasp representation**: some grasp database representations such as grasp densities (Detry et al., 2011) represent not a single successful grasp, but a complete continuous grasp success space which include the grasp neighborhood successes.

3. **Uncertainties in execution**: when executing a grasp from a grasp database the actual executed grasp will likely be a grasp from the neighborhood of the selected grasp due to pose uncertainties.

We investigate the transferability of the grasp neighborhood by randomly generating grasp configurations that are simulated in different environments. Comparing the outcome of hundreds of thousands of grasps simulated in different environments enable us to get insight into the nature of the transferability of a grasp database, which allow us to qualitatively show the importance of using context specific simulation when computing grasp databases for data-driven grasp planners. In addition, we will present a quality measure for the context transferability of a grasp database and use it to compute the transferability for several chosen objects and contexts.

In Section 2 we first discuss transferability and how we can measure it. Then in Section 3, we introduce related work within data-driven grasping and how it is currently used in the community. We introduce the setup: objects, grippers and control strategies in Section 4 which is followed by a description of how we compute the grasp affordances in Section 5. In Section 6 the results are presented and a detailed analysis is followed in Section 7.

### 2. Quantifying transferability

The transferability measure that we present in this section require a set of grasps that have been executed in multiple contexts. The binary outcome (success/failure) of these grasp experiments then define the transferability.

This is illustrated in Fig. 1 where the neighborhood (blue area) of successful grasp experiments in one context a is different from the neighborhood of successful grasp experiments from another context b. The green point illustrates the target grasp $g_s$ which belongs to the set of stable grasps $G_{stable}$ (black line/area) and the arrows illustrate three grasps $(g_1, g_2, g_3)$ chosen from the neighborhood of $g_s$. In context b grasps $(g_2, g_3)$ are no longer part of the neighborhood of successful grasps (indicated as red arrows) and therefore fail. We define the transfer of a grasp from one context to another to be successful in the case of $g_1$, but unsuccessful in the case of $(g_2, g_3)$.

If we think of transferability as a measure that determines how well we can predict outcomes of grasp experiments in one context, given the knowledge of the outcomes of the same experiments from another context, then we can borrow metrics from the machine learning literature and pose the measure of transferability as a classification problem. If we consider the execution of the same grasp but in two different contexts then we may have four outcomes: (success,success), (failure,success), (success,failure) and (failure,failure). These are illustrated in a confusion matrix in Table 1 as true positive (TP), false negative (TN), false positive (FP) and true negative (TP). In the above example $g_1$ will thus belong to TP and $g_2, g_3$ will belong to FP.

The probability that a selected successful grasp from $s_{float}$ will also be successful in $s_1$ is then $TP/(FP + TP)$.

We use the Matthews correlation coefficient (MCC) (Matthews, 1975) as the transferability measure. It is based solely on the confusion matrix and produces a value between $-1$ and $1$, where $1$ indicate perfect prediction, $0$ indicate a random prediction and $-1$ indicate an inverse prediction of a binary classification. It can be calculated directly from the confusion matrix as shown in (1).

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$ (1)
3. Data-driven grasping

State of the art grasp planning often rely on offline computed databases of feasible grasps generated using analytic, kinematic or dynamic simulation (Morales et al., 2006; Berenson et al., 2007; Goldfeder et al., 2009a,b; Goldfeder and Allen, 2011). These grasp databases are used in online grasp/motion planning where the best grasp is chosen to be used in a specific context. The main steps in the generation of a grasp-database and its usage is outlined in Fig. 2.

In the offline computation a sampling strategy is chosen to sample the parameter space of the system. This is at least defined by the pose of the gripper, the configuration of the gripper fingers (the grasp preshape) and the maximum joint torques. Additional parameters can also be used depending on the policy used to perform the actual grasping execution. The sampled parameters are used to initiate the grasp simulation which consist of the gripper and an object. When a single grasp has been generated the simulator is started and a grasping policy is used to control the fingers into a grasp. This policy is typically simple and often dependent on which type of simulation is used eg. kinematic or dynamic.

The simulation may be terminated due to a number of different criteria, which themselves involve the determination of whether the grasp was successful or not. In a kinematic simulation, analytic stability measures computed based on contact points are used to describe if a grasp is stable. These measures also apply to dynamic simulations, but here the stability might also be determined by performing a lifting motion with the gripper and object. If the object is still in the grasp after moving the gripper, then a stable grasp has been computed.

All successful grasps of a certain quality will be added to the grasp database. This quality might be based on a grasp wrench space (GWS) analysis (Ferrari and Canny, 1992), on tactile based quality metrics (Jørgensen and Petersen, 2010) or any other form of quality measure (Suárez et al., 2006). The main importance is to only store grasps of good quality.

For the experiments done in this paper no quality other than success and failure is used. These are defined by trying to move the object after it is grasped. If the grasp can withstand the accelerations during movement then the grasp is successful.

In the online computation the context is changed. We have a robot with a limited reach, we have obstacles that limits the grasping possibilities, sensors that are imperfect and other constraints imposed by the task that we wish the robot to perform. The main problem is then to select a grasp from the grasp database which is feasible in spite of these limitations and constraints. In (Berenson et al., 2007) they present a scoring function that favor grasp configurations that are farthest from obstacles.

Several learning approaches to grasping (Pelossof et al., 2004; Curtis et al., 2008) use simulated grasp databases as training data. The main goal is typically to use learning to infer a feasible grasp when presented with some form of input, which could be point cloud data, images, or simply the geometry of an object.

In (Goldfeder et al., 2009b) a grasp database “the Columbia Grasp Database” is presented and it is used for grasp planning in (Goldfeder et al., 2009a). The database is generated using dynamic simulation of grasps in a obstacle free environment and the grasps configurations where calculated by their eigen-based grasp planner.

In (Morales et al., 2006) the grasp database is
generated based on a kinematic simulation of closing the gripper fingers around the object in an obstacle free environment. The authors do not comment on the context independence of the database but point out that a gripper with tactile feedback is necessary to execute grasps generated with their approach. In (Berenson et al., 2007) the same approach is used to generate their grasp database, however they do not mention the use of tactile feedback in their experimental setup.

The use of tactile sensors to online guide the grasp stability has been investigated in several papers (Jørgensen and Petersen, 2008; Morales et al., 2007; Hsiao et al., 2010) and show great promise. However, the performance of tactile sensor hardware is far from ideal and typical issues are drift, low sensitivity range, low durability, detection of sliding, detection of measuring normal and measuring sheer forces. Reliable and general reactive grasping therefore still remain to be seen.

However, all above approaches rely on databases generated in a context independent simulation. To rely on the quality of those grasps it is necessary to have either a perfectly calibrated setup with no uncertainties or an online grasp execution that rely on tactile feedback to correct for uncertainties. Both of these properties in a setup are non-trivial and the latter is still not solved and is actively researched.

4. Objects and grippers

Three objects (see Fig. 3) were selected from the KIT Object-Models Web Database\(^1\). The objects are common household objects which are sufficiently different to provide interesting comparisons.

The grippers used to grasp these objects are the Schunk parallel gripper (PG 70) and the Schunk Dexterous Hand (SDH-2), which are both shown in Fig. 4. The PG 70 gripper has two fingers coupled into one Degree Of Freedom (DOF), that is, 1 DOF moves both fingers. The fingers can move up to 6.8 cm apart and the contact surface is approximately 2x3 cm and covered by rubber.

The SDH-2 is a 3-fingered dexterous hand with 2 DOF per finger and one coupled DOF to control the base rotation of two of the three fingers. The SDH-2 has 6 contacting surfaces covered with rubber, each of them measuring approximately 2x3 cm. However, for precision grasps only the 3 contact surfaces on the distal joints are normally used.

For the PG 70, we choose only one preshape as with the maximum distance of 6.8 cm between its jaws. Four preshapes were chosen for the SDH-2 which are shown in Fig. 5. These different preshapes enables different grasping options, and as such are important when characterizing the grasp affordances of the gripper.

We use preshapes because:

- Using preshapes is a simple and direct way of providing multiple ways of grasping an object. The grasping process reduces to the sequence: (open hand, move towards object, close hand), which is already supported by the software of the SDH-2.

- Preshapes do not require additional parametrisation. This property is important when sampling random grasps since the dimensionality of the search space is not extended by any gripper parameters such as the individual joint configurations.

\(^1\)http://wwwiaim.ira.uka.de/ObjectModels
5. Computation of Grasp Affordances

We compute complete grasp affordances by evaluating randomly sampled gripper poses in the neighborhood of an object. Each of these sampled poses, combined with a preshape of the gripper, represents a grasp hypothesis, which can be evaluated in simulation. The outcome of the simulation may be one of \{success, failure or collision\}, where success represents a successful grasp of the object (indicated by the fact that the fingers are still in contact with the object after grasping it) and failure represents a grasp where the gripper has no contact with the object after trying to grasp or lift it; collision represents a grasp, where the gripper is in collision with the object or the environment in the initial state of the simulation.

The evaluation of a grasp hypothesis is performed using a dynamics grasp simulator from RobWork (Jørgensen et al., 2010). The main simulation process of a single grasp is:

1. Set the initial scene configuration, eg. gravity, friction, poses of obstacles and objects.
2. Place the gripper in a sampled pose relative to the object and set the gripper configuration to one of the preshapes.
3. Test if the gripper collide with object or environment.
4. Start the simulation and set the target gripper configuration (preshape dependent) of the gripper controller.
5. If a grasp is obtained, lift the object and compute the success criteria.

We shall now discuss these steps in more detail.

5.1. Initial scene configuration

The initial scene configuration for generating the grasp-databases use a free floating environment, in which no gravity or obstacles are present. In such an environment the object can freely slide into or out of a grasp.

To investigate the transfer of grasp affordances of an object from the free floating environment we repeat the grasp simulation, but with the object placed on a table. Two canonical poses of each object are used to create two different table environments per object.

5.2. Object specific sampling strategy

The sampling of the gripper configuration and the choice of sampling strategy necessarily influence the resulting set of grasp affordances and the overall success probability. Typically, the primary goal of grasp planning is to maximize the grasp success probability by exploiting knowledge of gripper, object and environment. This tends to generate grasp databases that only represent a small subset of the complete set of successful grasps.

In this work, we need an approximation of the complete set of grasp affordances that are possible on a specific object. Thus, an unbiased sampling strategy that explores $SE(3)$ fully is preferred. Uniform random sampling in $SE(3)$ would therefore be ideal but it is also impractical because of the large number of simulations necessary to cover $SE(3)$. Instead a sampling strategy that is biased toward the object geometry is used. This effectively reduces the number of required simulations without reducing the success space too much.

Beside geometric models of the object the sampling also requires the approach vector of the gripper to be placed in the same direction as the positive $z$ axis of the gripper Tool Center Point (TCP) frame. The sampling effectively encapsulates the idea that the gripper needs to point toward some part of the object geometry before it is able to successfully grasp the object.

First a random point $p$ on the surface of the object is selected. Then an orientation $R$ is selected from a uniform distribution in $SO(3)$ and used to define the temporary target pose $(p, R)$. The pose is then translated along the $z$ axis by a randomly generated value $d$ in the interval $[-0.04\text{m};0.04\text{m}]$. The final pose is therefore:

$$T_{\text{pose}} = (p - (R \cdot [0,0,1]^T) \cdot d, R) \quad (2)$$
5.3. Collision filtering and labeling
For each sampled grasp configuration a collision detection between gripper and object/environment is performed before doing the actual simulation. If a grasp is colliding with the object then it will not be added to any database. If a grasp is colliding with the environment then it will be added and labeled Colliding.

In a grasp-planning context all colliding grasps will be left out of the database, but for this work we need to evaluate the transfer of success between grasp-databases and the colliding grasps will be needed to create meaningful statistics. That is any successful or failed grasp from database A that are labeled as colliding in database B should be left out when computing the transfer statistics between A and B.

5.4. Grasp simulation
The final step before adding a non-colliding grasp to the database is the grasp simulation. Both pose and configuration of the gripper has been sampled and no collisions with environment or object was detected. The simulation is initialized with the scene configuration and the sampled parameters. When the simulation starts, a penalty based grasp controller guides the fingers toward a closing configuration. Closing configurations for the four preshapes of the SDH-2 is shown in Fig. 5. A grasp is successful if the object gets caught between the fingers and if it stays their under a 10 cm movement in the positive z-axis of the world coordinate frame (gravity works in the direction of the negative z-axis). This is termed a lifting operation, which only makes sense in the table environments where the objects needs to be lifted free of the table.

6. Simulation Results
We provide simulation results in terms of complete sets of affordances for 3 objects (see Fig. 3) using the PG 70 parallel gripper and SDH-2 hand in both of the two different contexts, namely the rather artificial free-floating context shown in Fig. 6 and the more application-oriented context where the object is placed on a table (see Fig. 6 b.c.e.f.h.i). The table environments have a gravity of $9.81 \text{m/s}^2$ and the viscous Coulomb friction between table and object is set to $0.3 \text{N/(m/s)}$. All simulated environments for all objects are illustrated in Fig. 6.

Fig. 7 illustrates the successful outcomes of the same 5000 grasp hypotheses in the three different contexts of the corny object grasped with the SDH-2 using the preshape $c_{par}$. It is clear that the added table constraint significantly reduces the number of successful grasps. However, it is not clear if successes from the constrained environments (center and right image) will also be successes in the floating environment. In the following, we use the confusion matrices introduced in Section 2 to evaluate how well successes and failures in the floating simulations transfer to successes and failures in the table environments, and vice versa.

Multiple datasets were generated for each gripper. For the SDH-2 the datasets are characterized by a triple $(o_i, s_j, c_k)$, where $o_i$ is the object, $s_j$ is the specific scene (free floating or on table with different poses), and $c_k$ is the grasp strategy, which includes the number of fingers and the preshape used. The parallel gripper is simpler, and only one grasp strategy is used. Hence we describe datasets generated with the PG 70 by a pair $(o_i, s_j)$.

For each floating environment experiment $(o_i, s_{\text{float}}, c_k)$ 100,000 grasp simulations were generated using the sampling approach presented in Section 5.2. The overall success probabilities of the simulations are available in Table 2. These indicate the size of the success space of the grasp datasets in the individual contexts. Two success probabilities are given. The first shows the percentage of grasp successes from all grasps that were not initially in collision, the second shows the percentage of grasp successes from all grasps including the colliding ones.

In the same table the success probabilities of the simulations of the table environments $(s_{\text{side}}, s_{\text{uprigt}})$ are also shown. These simulations use the same grasp hypotheses as in the floating environment, but performed in the specific contexts (e.g. the edge of the table).

That the exact same grasps have been executed in the different contexts enables us to calculate the transferability measure presented in Section 2. Table 3 shows the confusion matrices from which the transferability is calculated and the transferability quality of each confusion matrix is presented in Table 4.

7. Analysis of results
The MCC metric is used directly on the confusion matrices of Table 3 (see Table 1 for explanation). Note

\[\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}\]
Tab. 2. Success percentages of the simulated outcomes. In each major column, the left subcolumn shows the percentages of success if collisions are not included, and the right column shows the success percentages if collisions are included.

<table>
<thead>
<tr>
<th></th>
<th>SDH-2</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>PG 70</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$c_{par}$</td>
<td>$c_{parsmall}$</td>
<td>$c_{ball}$</td>
<td>$c_{cy}$</td>
<td>$c_{0}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$s_{float}$</td>
<td>43.1% 25.0%</td>
<td>45.5% 5.2%</td>
<td>70.3% 38.0%</td>
<td>52.9% 28.0%</td>
<td>12.3% 0.7%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$s_{side}$</td>
<td>5.8% 0.4%</td>
<td>0.0% 0.0%</td>
<td>5.8% 0.2%</td>
<td>5.3% 0.3%</td>
<td>0.1% 0.0%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$s_{upright}$</td>
<td>42.1% 7.2%</td>
<td>41.5% 1.9%</td>
<td>70.9% 9.3%</td>
<td>51.9% 8.9%</td>
<td>14.4% 0.4%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$s_{float}$</td>
<td>54.5% 42.6%</td>
<td>47.7% 6.0%</td>
<td>79.8% 61.1%</td>
<td>61.7% 45.7%</td>
<td>41.5% 3.7%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$s_{side}$</td>
<td>47.0% 5.8%</td>
<td>48.4% 1.7%</td>
<td>72.5% 5.4%</td>
<td>47.3% 6.0%</td>
<td>44.0% 1.6%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$s_{upright}$</td>
<td>45.0% 4.7%</td>
<td>65.4% 2.3%</td>
<td>81.9% 4.9%</td>
<td>47.7% 4.9%</td>
<td>70.4% 3.5%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$s_{float}$</td>
<td>48.5% 24.9%</td>
<td>30.4% 5.5%</td>
<td>84.9% 72.0%</td>
<td>72.3% 59.9%</td>
<td>3.8% 0.3%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$s_{side}$</td>
<td>22.7% 2.3%</td>
<td>10.9% 0.3%</td>
<td>34.4% 1.9%</td>
<td>32.8% 3.4%</td>
<td>2.4% 0.1%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$s_{upright}$</td>
<td>43.4% 6.6%</td>
<td>23.1% 1.4%</td>
<td>75.3% 7.5%</td>
<td>63.1% 10.4%</td>
<td>17.2% 0.6%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Tab. 3. Confusion matrices of successes and failures from floating environment and the specific table environment ($s_{side}$/$s_{upright}$). See table 1 for a detailed explanation of a single cell.

<table>
<thead>
<tr>
<th></th>
<th>SDH-2</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>PG 70</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$c_{par}$</td>
<td>$c_{parsmall}$</td>
<td>$c_{ball}$</td>
<td>$c_{cy}$</td>
<td>$c_{0}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$s_{side}$</td>
<td>159 221</td>
<td>0 0</td>
<td>74 113</td>
<td>97 222</td>
<td>0 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$s_{upright}$</td>
<td>6271 861</td>
<td>1777 100</td>
<td>8586 761</td>
<td>6296 830</td>
<td>327 46</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$s_{side}$</td>
<td>1660 8183</td>
<td>284 2379</td>
<td>1127 2712</td>
<td>1587 5017</td>
<td>34 2182</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$s_{upright}$</td>
<td>4876 916</td>
<td>1548 126</td>
<td>1459 407</td>
<td>893 314</td>
<td>1568 81</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$s_{side}$</td>
<td>1903 4621</td>
<td>241 1546</td>
<td>303 406</td>
<td>474 871</td>
<td>175 1924</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$s_{upright}$</td>
<td>1683 4059</td>
<td>155 1061</td>
<td>195 297</td>
<td>449 1146</td>
<td>78 1399</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$s_{side}$</td>
<td>1019 228</td>
<td>246 31</td>
<td>445 112</td>
<td>1063 278</td>
<td>30 35</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$s_{upright}$</td>
<td>1310 2923</td>
<td>214 2045</td>
<td>570 494</td>
<td>1212 1531</td>
<td>43 2568</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$s_{side}$</td>
<td>1518 119</td>
<td>547 82</td>
<td>2167 91</td>
<td>2019 112</td>
<td>113 434</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$s_{upright}$</td>
<td>538 1591</td>
<td>219 1874</td>
<td>261 477</td>
<td>607 602</td>
<td>53 2584</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
that in Table 3 collisions to the table are filtered away. The results are shown in the rows of Table 4.

The results show that the quality of the prediction depends highly on both the gripper and the object/context that is used. In about half the contexts the MCC is about 0.80, indicating a fairly good transfer. In the other half the quality ranges from 0.1 to 0.5.

It is important to mention that in certain contexts quite a number of grasps that are not successful in the free-floating environment are successful in the constrained environment. This implies that in certain scenarios context-specific control strategies need to be taken into account. For example, the movement of a flat object on a table when touched by one of the fingers might be utilized in the grasping process. These context-specific constraints cannot be accounted for in a free-floating scenario and need to be learned in the specific context.

In summary, the results show that grasp simulation in free-floating scenarios give in general fairly unreliable indications of grasp success in more constrained scenarios. Furthermore, in certain contexts \((C_{par, O_{corny, side}}, C_{par, O_{corny, side}})\) and \((C_{ball, O_{corny, side}})\) the prediction is so bad that the chances of choosing a successful grasp from \(s_{float}\) which will also be successful in \(s_{1}\) is less than 20\%. This is calculated by the relationship between TP and FP values of the confusion matrices in Table 3. Hence, grasp databases should ideally be learned in a context-specific fashion.

### 8. Conclusion

Data-driven grasp planning has become increasingly popular and several grasp databases are available for public download. In this paper we showed that it is likely that grasps performed on objects in different contexts have different outcomes. Hence, one should take care when using a grasp database in a context other than the context in which it was generated.

We specifically investigated the transfer between an unconstrained, free-floating environment and more constrained environments, which is important for applying learned grasp knowledge in novel contexts. We showed that grasp success likelihoods depend strongly on the context the object is embedded in.


