PALADYN Journal of Behavioral Robotics

VERSITA

Research Article · DOI: 10.2478/s13230-012-0020-5 · JBR · 3(2) · 2012 · 54-62

VisGraB: A Benchmark for Vision-Based Grasping

Gert Kootstra[1]*, **Mila Popović**[2], **Jimmy Alison Jørgensen**[3], **Danica Kragic**[1], **Henrik Gordon Petersen**[3], **Norbert Krüger**[2]

1 Computer Vision and Active Perception Lab, CSC, Royal Institute of Technology (KTH), Stockholm, Sweden

2 Cognitive Vision Lab, The Mærsk Mc-Kinney Møller Institute, University of Southern Denmark, Campusvej 55, DK-5230 Odense, Denmark

3 Robotics Lab, The Mærsk Mc-Kinney Møller Institute, University of Southern Denmark, Campusvej 55, DK-5230 Odense, Denmark

Received 2011/12/10

Accepted 2012/04/28

Abstract

We present a database and a software tool, VisGraB, for benchmarking of methods for vision-based grasping of unknown objects with no prior object knowledge. The benchmark is a combined real-world and simulated experimental setup. Stereo images of real scenes containing several objects in different configurations are included in the database. The user needs to provide a method for grasp generation based on the real visual input. The grasps are then planned, executed, and evaluated by the provided grasp simulator where several grasp-quality measures are used for evaluation. This setup has the advantage that a large number of grasps can be executed and evaluated while dealing with dynamics and the noise and uncertainty present in the real world images. VisGraB enables a fair comparison among different grasping methods. The user furthermore does not need to deal with robot hardware, focusing on the vision methods instead. As a baseline, benchmark results of our grasp strategy are included.

Keywords

grasping of unknown objects · vision-based grasping · benchmark

1. Introduction

Grasping previously unseen objects based on visual input is a challenging problem. Various methods have been proposed for solving the problem, as will be discussed later, but it is difficult to compare them and evaluate their strengths and weaknesses. This is due to the fact that methods are often tested on different data and with different hardware setups in different labs, which makes it difficult, if not impossible, to repeat the experiments under the same conditions. It is furthermore difficult to quantify results thoroughly, because of the time consuming nature of the experiments. For these reasons, we propose a mixed real-world and simulated benchmark framework.

A database of stereo images is provided and the generated grasps are evaluated using a simulated environment, [8, 15], see Figure 1. This setup allows for extensive experimental evaluation, supporting comparison of different methods, while considering noise and uncertainty in the real stereo images. Our previous work used a part of the database as a proof of concept, [27]. In this paper, we present a large database along with software tools to evaluate the generated grasps.

The proposed benchmark focuses on grasping unknown objects in realistic, everyday environments without prior knowledge. The grasp-generation methods have to deal with the fact that the visual observation provides only partial and noisy information of the scene and that no prior object models are available. This poses a challenging but important problem that needs to be solved if to advance in autonomous robotics. The problem is currently actively studied in the robotics community and different methods have been proposed. For instance, to deal with the noisy and incomplete data coming from robotic sensors and to provide a reduced set of potential grasps, shape approximations using shape primitives have been used in [11, 14]. A less restricted strategy for grasping unknown objects in the real world based on a hierarchical edge representation of the scene has been presented in [28]. In [27], this method has been extended to include surface information. Other approached apply learning methods to gain grasp experience and apply this in grasping unknown objects, for instance, based on the parameters of a superquadric representation of the object [7, 26], shape context [3], or features of edge elements [2]. Training can be performed on simple geometrical shapes [6], synthesized objects [31], or using human expertise [7]. In [12], a publicly-available database with a large number of performed grasps has been created, which can be used to train machine learning algorithms for grasping novel objects.

The presented database contains original stereo images, where no object hypotheses are generated beforehand. This means that the grasp-generation methods provided by the users of the benchmark need to be able not only to deal with the grasp-generation process but also with generating object hypotheses, if the grasp generation method requires that. Methods such as [27, 28, 31], works directly on images without the need to explicitly generate object hypotheses. There are also several methods that perform figure-ground segmentation at first, such as using a bottom-up segmentation method based on color and depth [30] and the additional use of a table plane detection [3]. Other

*E-mail: kootstra@kth.se

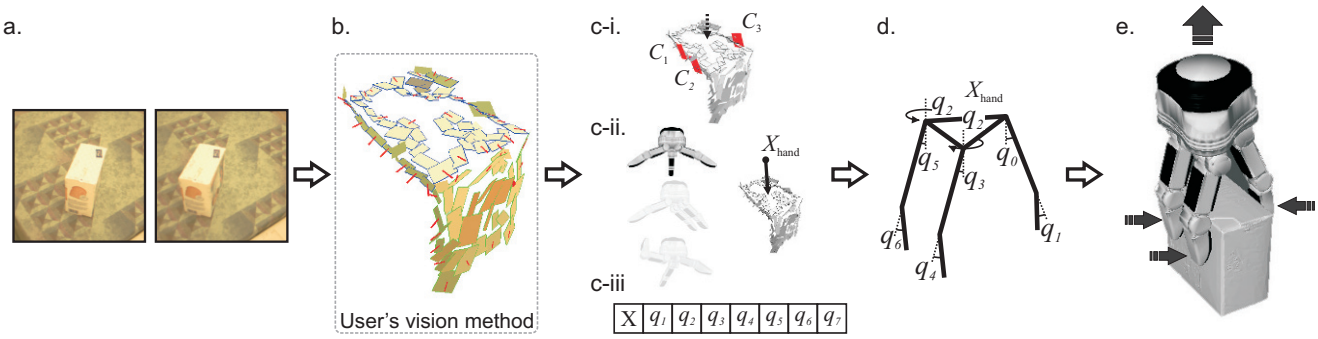

Figure 1. *The benchmark pipeline. The real stereo images (a) are input to the user's grasp-generation method (b). Our method is given as a baseline example. The grasp-generation method proposes a grasp hypothesis, either as (c-i) a set of desired contacts , $C = \{C_1, C_2, C_3\}$ (implicitly coding the approach side), (c-ii) by choosing one of the hand pre-grasps and the desired hand pose, or (c-iii) by directly setting the joint angles and hand poser. Based on the grasp hypothesis, the hand pose, $\mathbf{X}_{\text{hand}}$, and configuration, $q = \{q_0, \ldots, q_6\}$, are determined by the provided software (d), and the grasp is executed by the dynamic simulator (e). Note that b) shows the object representation specific to our baseline method (see Section 4).*

methods do not need a segmentation of the scene, because they use single image points for pinch grasps, e.g., [31].

Although the studies discussed all deal with grasping unknown objects, each used a unique experimental setup; different objects and scenes were used, as well as different robotic platforms. Some of the studies have been done entirely in simulation (e.g., [6, 7, 12, 26]), whereas others are performed in the real world (e.g., [2, 28, 31]). Moreover, the studies use different measures to evaluate the grasp performance; [12, 14, 26] used a measure based on the grasp wrench space [10], the time to generate the first good grasp is used in [6], and other studies grasp and lift the object in order to determine grasp success [27, 30, 31], or a more fine-grained grasp classification [2, 28]. We use all these quality measures in our benchmark.

In this paper, we propose VisGraB as a standardized benchmark for grasping unknown objects based on real-world visual data. We test the performance of the grasp-generation method by executing the grasps in a dynamic simulator, lifting the object, and evaluating the result using different quality measures: success rate, fine-grained grasp classification, time to successful grasp, and a measure based on the grasp wrench space. By enabling the comparison between different grasping methods, we aim to provide a better insight into the different methodologies and their outcomes.

We see the use of a grasp simulator as a good solution to obtain a standardized comparison between methods. Although a grasp simulation on an individual grasp level will not be completely identical to reality due to the inherent complex nature of the physical processes, methods are likely to be ranked correctly on a more general level. This is supported by our recent work [21], where we tested the same methods using VisGraB and two real robotic setups. Methods that tested successful in simulation performed well in reality and reversely, poor performing methods in simulation performed poorly in reality as well. We therefore believe that the simulation is a valid tool to evaluate grasping methods. Furthermore, in [8], thousands of grasps with a parallel griper have been compared between our simulator and a real system. Simulator and reality agreed on the clearly stable and clearly unstable grasps. Differences were found for just-stable and just-unstable grasps. However, we do not see this as a problem, since we aim to aid the development of robustly stable grasping method.

In other fields, benchmarking is quite common, for instance, for object categorization and image segmentation [9, 13], for stereo-correspondence algorithms [32], and for validation of 3D-reconstruction methods [33]. The wish for a standardized test for grasping has also been put forward in [34], where a benchmark is presented for the evaluation of grasp planners. However, different from our aims, the benchmark in [34] focuses on grasping known objects based on full and detailed geometrical information about the objects. We, on the other hand, propose a benchmark for grasping unknown objects in complex scenes based on real, incomplete, and noisy visual observations.

In summary, the main contribution of this paper is a standardized benchmark for vision-based grasping of unknown objects, so that different grasp generation methods can be systematically tested and compared. VisGraB includes: 1) A database with real stereo images and simulated models of a large number of scenes containing objects to be grasped, 2) software for the easy access of the database and use of the simulator, 3) the execution of the grasp hypotheses in a dynamic simulation, 4) an evaluation of the grasps based on static and dynamic quality measures, and 5) tools to display the results. Using this benchmark allows users to focus on the vision aspects of grasping, without having to deal with the robotic hardware.

The paper is organized as follows: We first describe the benchmark with the database, the dynamic simulator, and the grasp quality measures in Section 2. In Section 3, a description of how to use the benchmark is given. Next, in Section 4, we give a baseline performance for the benchmark using our method described in [27]. The paper ends with a discussion in Section 5.

2. The Benchmark

The benchmark consists of a database containing real visual input, a grasp simulator including a dynamics engine to evaluate the grasps, and several software tools for easy access to the database and use of the simulator, as well as evaluation and presentation of the results. The benchmark contains a total of 432 scenes with a variety of different objects and with different backgrounds. The database includes real stereo images of all the scenes, as well as the 3D models of the scenes, which will be used by the simulator to evaluate the grasps.

The general pipeline of the benchmark is illustrated in Figure 1. Based on the stereo images (Fig. 1a), the user's method generates grasping hypotheses (Fig. 1b). The hypotheses can be provided in different formats (Fig. 1c). Given a grasping hypothesis, the software provided

Figure 2. *The 18 objects used in the benchmark.*

with the benchmark determines the pose of the hand and the joint configuration (Fig. 1d). The grasp is then executed by the simulator and the quality of the grasp is displayed to the user (Fig. 1e). Details on the database are given in Section 2.1. Section 2.2 describes the grasp simulator, and the possible grasp representation are given in Section 2.3. Finally, Section 2.4 describes the evaluation of the grasps.

The benchmark, including stereo images, the modeled 3D scenes, and the simulation software can be found on the VisGraB website [20].

2.1. The Database

The 18 objects used in the database are displayed in Figure 2. The objects are part of the KIT ObjectModels Web Database[1]. 3D models of all objects are available for the grasp simulation. The objects have various shapes, sizes, colors, and textures. We recorded scenes with one object and with two objects. In the single-object case, we recorded the 18 different objects in eight different poses, four where the object stands upright, and four where the object lies down. In the double-object scenes, we have 9 combinations of objects, where the objects are in eight different configurations, four where the objects are placed apart, and four where the objects touch each other. All scenes are recorded in two conditions, placed on a non-textured and on a cluttered/textured table. This gives in total $2\times(18\times8 + 9\times8) = 432$ scenes. Some example scenes are given in Figure 3, top row.

The scenes are modeled in 3D, in order to test the user-generated grasps in simulation. The models are obtained by calculating the 3D point cloud of the scene using the dense stereo algorithm provided in OpenCV, and subsequently registering the 3D object models to the point cloud using rigid point-set registration [25]. Where necessary, the registration was corrected by hand. A few scene models are shown in Figure 3, bottom row.

The object models, taken from the KIT ObjectModels Web Database, have been scanned using a laser-range finder and are of high quality, with sub-millimeter errors. Errors in the positioning of the objects and the table in the scene are in the order of a few millimeters.

With the database, the vision-based grasping methods are tested for the ability to generate grasps on objects with a variety of different shapes, sizes, colors and textures. Furthermore, the robustness to the pose of the object, the complexity of the scene and the clutter in the scene is tested.

2.2. The Grasping Simulator

The grasps are performed in simulation using RobWork[2], see Figure 1. RobWork is a framework for simulation and control of robot systems [15, 16, 24], with a special emphasis on object grasping and manipulation [5, 17, 18]. The grasp simulator has been evaluated and compared to real systems in [4, 8] and has been used, for instance, in [1, 19, 27]. For the dynamics simulation and constraint solving, RobWork relies on Open Dynamic Engine (ODE), one of the most used physics engines for robotics. In addition, RobWork performs its own, more accurate, contact calculation for improved grasp simulation. We provide the RobWork grasp simulator as a part of the VisGraB benchmark and created an easy-to-use interface, which allows the user to work with VisGraB without having to learn the details of the simulator. Our benchmark methodology is based on open xml formats and can hence be used with other grasp simulators, such as GraspIt [23] and OpenGRASP [22]. However, as motivated above, RobWork provides a good grasp simulation that has the additional benefit that it is developed by us, allowing good integration in VisGraB and swift application of updates and improvements. RobWork is distributed under the Apache 2.0 license and is supported on both Windows and Linux-based operating systems.

Using the RobWork grasp simulator including a dynamics engine allows us to not only look at static quality measures of the grasp, but also to determine the actual grasp success by observing the dynamical and physical consequences of the grasp. In our definition, a stable grasp is a grasp with which the object can be lifted without slipping from the hand. We therefore propose a method where the object is lifted after it has been grasped. We hence define the lift-quality measure as an important measure for the stability of a grasp, but also provide a static quality measure based on the grasp wrench space. The quality measures are explained in Section 2.4.1.

We use the three-finger Schunk Dexterous Hand (SDH) (see Figure 4), which can be used for both two-finger parallel and three-finger grasping. The SDH has seven degrees of freedom, allowing for complex and flexible grasping. We denote the joint configuration as $\mathbf{q} = \{q_0, \ldots, q_6\}$. Although we made the decision to use the SDH, RobWork supports the easy use of other grippers.

The objects in the scene are modeled as rigid bodies, and are not deformable. They all are assumed to have the same friction properties. The simulation uses a Coloumb friction approximation, with the following friction coefficients: $\mu = 0.6$ for object-finger contact, $\mu = 0.8$ for finger-finger contact, and $\mu = 0.4$ for object-table contact.

A grasp is performed by first placing the hand in a suitable grasp configuration generated by the user's vision algorithm and using the utility functions provided by the benchmark, see Section 2.3.2 and 2.3.1. The simulation is then started and a grasp-control policy guides the fingers from the start configuration $\mathbf{q}_{\text{open}}$ towards the closed configuration $\mathbf{q}_{\text{closed}}$. When the fingers achieve a static configuration, it is either because of contact forces or because $\mathbf{q}_{\text{closed}}$ is reached. Next, the system attempts to lift the grasped object. After lifting, the quality of the grasp is determined as explained in Section 2.4.1.

The grasp control policy is fairly simple, but can directly be used on the interface of the real hardware of the SDH as well. The policy does

[1] http://wwwiaim.ira.uka.de/ObjectModels

[2] http://www.robwork.dk

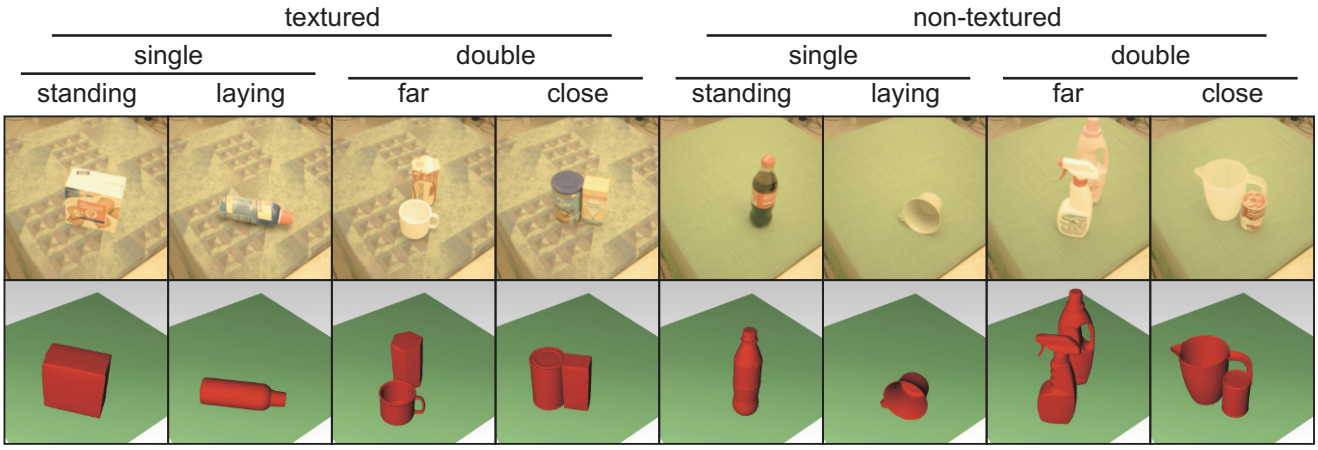

Figure 3. *Examples of scenes included in the database. The top row gives the rectified left camera images and the bottom row gives a view on the modeled scenes used for grasp simulation. Examples of the different conditions are given.*

not rely on specific sensor feedback other than the joint angles. It requires two joint configurations of the hand $\mathbf{q}_{\text{open}}$ and $\mathbf{q}_{\text{closed}}$, as well as the maximum allowed joint torques τ_{max}. The user moreover needs to provide $\mathbf{X}_{\text{hand}}$, which is the 6-dimensional Cartesian pose of the hand base (position and orientation in 3D). The control policy will close the fingers from $\mathbf{q}_{\text{open}}$ toward $\mathbf{q}_{\text{closed}}$ using a PD controller on each joint. The torque used by the PD controller will be limited by τ_{max} which allows for a rough balancing of the contact forces. As such the simulation only need a few parameters to execute a grasp:

$$(\mathbf{X}_{\text{hand}}, \mathbf{q}_{\text{open}}, \mathbf{q}_{\text{closed}}, \tau_{\text{max}}) \tag{1}$$

These parameters make out the grasp configuration and should be the output of the grasping strategy that is being benchmarked. However, many vision-based grasp strategies do not include grasp control specifics such as inverse kinematics or explicit modeling of joint force limits. To accommodate the need for varying levels of grasp control, the benchmark provides two utility functions that ease the generation of grasp configurations, which are outlined in the next section.

2.3. Grasp Utility Functions

To simplify the generation of grasps, we provide three grasp utility functions as part of the benchmark: based on grasp contacts (Fig. 1c-i), based on hand pre-shapes (Fig. 1c-ii), and based on the the joint configuration (Fig. 1c-iii).

2.3.1. Grasp contacts

The grasp parameters can also be generated by providing two or three desired grasp contacts. See Figure 1c-i for an example of three contacts. A contact $\mathbf{C}_i = \{\mathbf{c}_{\text{pos}}, \mathbf{c}_{\text{dir}}\}$ indicates the position, $\mathbf{c}_{\text{pos}} = \{c_x, c_y, c_z\}$, where the tip of the finger should be placed and the contact direction, $\mathbf{c}_{\text{dir}} = \{c_{d1}, c_{d2}, c_{d3}\}$, which determines in which direction the contact force should work. The inverse kinematics are solved by the utility function provided in the benchmark:

$$\mathbf{C} \mapsto (\mathbf{X}_{\text{hand}}, \mathbf{q}_{\text{open}}, \mathbf{q}_{\text{closed}}, \tau_{\text{max}}) \tag{2}$$

where $\mathbf{C} = \{\mathbf{C}_1, \mathbf{C}_2\}$ for two-finger grasps and $\mathbf{C} = \{\mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3\}$ for three-finger grasps.
The inverse kinematics algorithm does not require the grasp contacts to be in a specific order or even to be part of the inverse kinematics solution. In the latter case, the algorithm generates inverse-kinematics solutions that are close to the desired configuration. However, configurations with too high deviation from the target configuration are reported as failed grasps.

2.3.2. Hand pre-shape

It is common to use hand pre-shapes in grasp planning, where the pre-shapes are either generated using simple heuristics or by expert users. For the SDH we have chosen three general hand pre-shapes, see Figure 4. The figure shows the opening and closing positions. The *2-finger parallel grip* is shown in left left column, the *3-finger ball grip* in the middle column, and the *3-finger cylinder grip* in the last column. Given the desired pose of the hand base, $\mathbf{X}_{\text{hand}}$ and the identifier for the specific hand pre-shape, k, the utility function calculates the grasp parameters:

$$(\mathbf{X}_{\text{hand}}, k) \mapsto (\mathbf{X}_{\text{hand}}, \mathbf{q}_{\text{open}}, \mathbf{q}_{\text{closed}}, \tau_{\text{max}}) \tag{3}$$

The complete description of the pre-shape configurations including τ_{max} is available on the VisGraB website [20].

2.3.3. Joint configuration

The user can also use his or her own inverse-kinematic solver to acquire the hand pose, $\mathbf{X}_{\text{hand}}$, and joint configuration when the fingers are in contact with the object, $\mathbf{q}$. The simulation parameters are then obtained with the utility function:

$$(\mathbf{X}_{\text{hand}}, \mathbf{q}) \mapsto (\mathbf{X}_{\text{hand}}, \mathbf{q}_{\text{open}}, \mathbf{q}_{\text{closed}}, \tau_{\text{max}}) \tag{4}$$

2.4. Experimental Evaluation

To test the quality of the user's grasp-generation method, we apply the following experimental procedure: The user provides a list of grasp

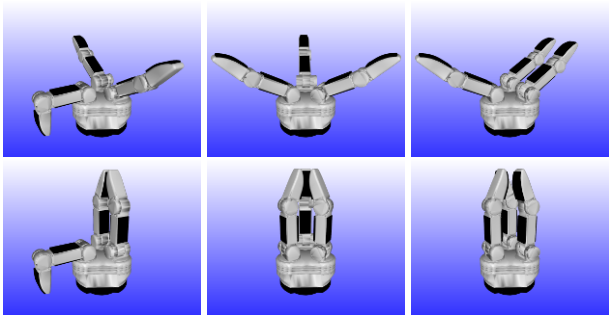

Figure 4. The hand pre-shapes. The top row depicts the $\mathbf{q}_{\text{open}}$ configurations and the bottom row the $\mathbf{q}_{\text{closed}}$ configurations. From the left to the right: 2-finger parallel grasp, 3-finger ball grasp and 3-finger cylinder grasp.

configuration for every scene in the database. All grasps are then performed by the simulator and the results are returned.
In a single experimental trial, the quality of the generated grasp is tested as follows: the hand is placed in the correct pose, $\mathbf{X}_{\text{hand}}$. It then closes from the opening configuration, $\mathbf{q}_{\text{open}}$, to the closing configuration, $\mathbf{q}_{closed}$. The object is grasped when the hand settles in a stable configuration and the fingers touch the object. However, this does not necessary mean that the grasp is stable. To test the stability of the grasp, the hand attempts to lift the object. We discriminate the following results:

Stable grasp: The object was grasped and held after lifting, with little or no slippage of the object in the hand.

Object slipped: The object was grasped and held after lifting, but there was considerable slippage of the object in the hand.

Object dropped: The object was grasped, but after lifting, the object was no longer held by the hand.

Object missed: The object was not grasped by the hand.

In collision: The initial hand configuration produced a situation where the hand was penetrating the object(s) and/or the table.

Invalid grasp contacts: The inverse-kinematics solver could not find a joint configuration to reach the desired grasp contacts.

Simulation failure: The simulation failed due to physics-engine failure.

We consider the grasp to be successful when the result is either *object slipped* or *stable grasp*. In both cases, the object is in the hand after lifting. The two situations are discriminated based on the amount that the object slipped in the hand during lifting. The slippage defines the lift-quality measure, In case of the double-object scenes, the results are given for the object that is closest to the hand.

2.4.1. Grasp quality measures

In case the object is lifted successfully, we calculate the grasp quality using two quality measures: the *lift quality*, Q_{lift}, and the *grasp wrench-space quality*, Q_{gws}.
The lift quality is a dynamic quality measure that represents the ability of a grasp to hold the object stable during lifting, that is, with the object slipping from the hand as little as possible. The lift quality is a value

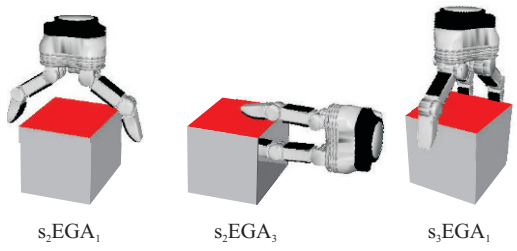

Figure 5. Illustration of the surface-based Elementary Grasping Actions used by the benchmark method [27]. The grasps are targeted at the red surface. s_2EGA_1 is an two-finger encompassing grasp, s_2EGA_3 is a two-finger side pinch grasp, and s_3EGA_1 is a three-finger encompassing grasp.

between 0.0 and 1.0 and it is inversely proportional to how much the object moves with respect to the hand during lifting:

$$Q_{\text{lift}} = 1 - \frac{||\mathbf{h} - \mathbf{o}||}{||\mathbf{h}||} \tag{5}$$

where $\mathbf{h}$ is the 3D displacement of the hand during lifting and $\mathbf{o}$ is the 3D displacement of the object during lifting.
The grasp wrench-space measure Q_{gws} is a static quality measure based upon the grasp wrench space (GWS), which reflects the minimum perturbating wrench that the grasp can counterbalance, given the forces of the fingers and the Coulomb friction coefficients [10, 23]. The GWS is determined by the friction cones of all n contact points. For a given contact i, the direction of the friction cone is determined by the contact force $\mathbf{f}_i$, and the width of the cone is based on the Coulomb friction coefficient, μ. To calculate the GWS, the cone is approximated by a set of m force vectors, $\mathbf{f}_{i,j}$, which are equally spread around the surface of the cone. For each force vector, a six-dimensional contact boundary wrench is defined as:

$$\mathbf{w}_{i,j} = \begin{pmatrix} \mathbf{f}_{i,j} \\ \frac{1}{r} \cdot \mathbf{d}_i \times \mathbf{f}_{i,j} \end{pmatrix} \tag{6}$$

where $\mathbf{d}_i$ is the vector from the torque origin to the ith point of contact and r is the maximum radius of the object from the torque origin. The cross product $d_i \times \mathbf{f}_{i,j}$ is the torque $\tau_{i,j}$. The GWS is then computed as the convex hull over the union of each set of contact boundary wrenches:

$$W = \text{ConvexHull}\left(\bigcup_{i=1}^{n}(\mathbf{w}_{i,1}, \ldots, \mathbf{w}_{i,m})\right) \tag{7}$$

Finally, the grasp quality measure Q_{gws} is determined by the distance from the origin to the nearest facet of the convex hull, which reflects the maximum perturbating wrench that the grasp can counterbalance.

2.4.2. Analyses and presentation of results

Since different grasping methods may have their own strengths and weaknesses, we do not summarize the results in a single value. Instead, we analyse the data in different ways. First, we give the distribution of grasping results for the different conditions, see Figure 6.

Second, we give the average grasp quality measures Q_{lift} and Q_{gws} over the successful grasps, i.e., *stable grasps* and *object slipped*, see Table 1.
These two analyses give the average performance, which indicates how well the method is expected to perform if one grasp of the suggested hypotheses is selected. However, grasp performance can be greatly improved if the system is allowed to attempt multiple grasps. To investigate this, we plot the grasp success rate as a function of the number of grasp attempts as a third analysis, see Figure 7. Here, per scene, grasps are selected at random from the list of hypotheses and the averages over the different scenes and 20 randomized trials are given. In the forth analysis, we investigate how many attempts are needed to achieve a successful grasp, see Table 2. This table gives the proportion of scenes where the method provides a successful grasp, and, if this is the case, how many grasp attempts are on average needed to grasp the object successfully.
Finally, to get more insight in the performance of the method for the different objects, we give the percentage of successful grasps for each object in the different conditions, see Tables 3 and 4.
Scripts are provided as part of the benchmark to process the results and to present the results.

3. Using the Benchmark

VisGraB is easy to use. The user does not need to learn to work with the grasp simulator, as this is all taken care of by the provided software. Tools are available to access the database, execute the grasps, evaluate the outcome and display the results. The only thing the user needs to add is his or her vision-based grasp-generation method, which takes the images as input and that suggests a list grasp hypotheses as output.
The benchmark can be downloaded from the VisGraB website [20]. Using the benchmark works in a number of steps:

1. Loading the stereo images and the stereo-calibration file.
2. Generating grasps based on the visual information and providing the grasp configurations, potentially by using the utility functions for hand pre-shapes or grasp contacts.
3. Running the simulation, providing a list of grasp configurations for every scene.
4. Running the scripts to process and represent the results.

The final benchmark results can then be published on the VisGraB website for comparison. The detailed information about the formats and the use of the software can be found on the website.

4. Baseline Method

To set a baseline for comparison and to illustrate the analyses, we used our grasp-generation method presented in [27] and applied it to the VisGraB benchmark. The grasping method is based on an Early Cognitive Vision system [29] that builds a sparse hierarchical representation based on edge and texture information. This representation is used to generate edge-based and surface-based grasps. The method detects surfaces of the objects in the scene, and generates grasps based on these surfaces. For the baseline, we use the surface-based grasps only. The grasp method finds contact points at the boundary of a surface, on which so-called Elementary Grasp Actions are applied, see Figure 5. Based on two grasp contacts, a two-finger encompassing grasp, $s_2\text{EGA}_1$, is generated, as well as two two-finger pinch grasps, $s_2\text{EGA}_3$ one for each contact. Based on three grasp contacts, a three-finger encompassing grasp, $s_3\text{EGA}_1$, is generated. For details about the method, we refer to [27].

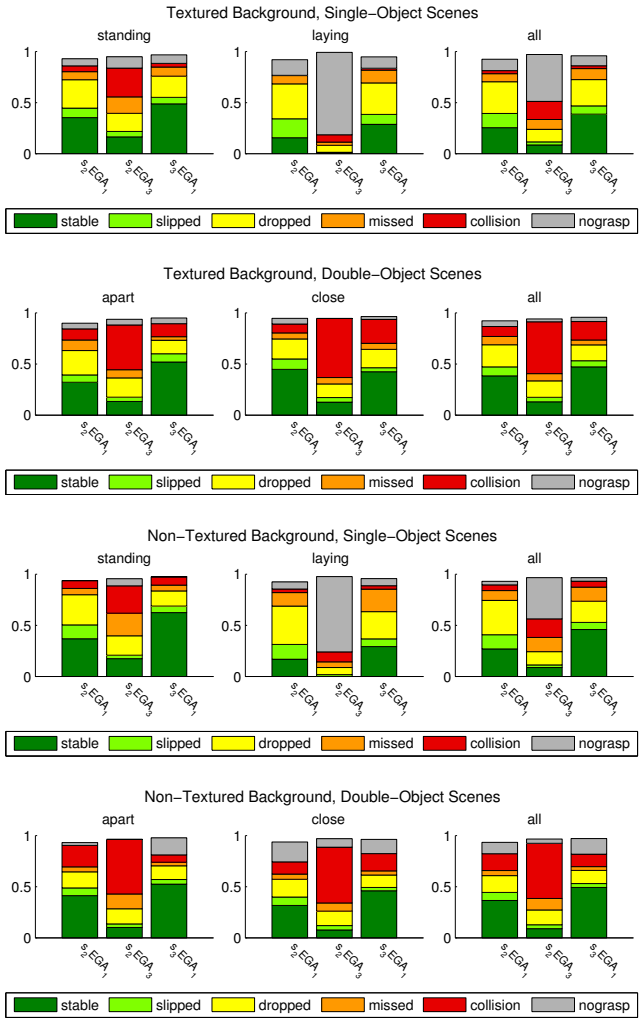

Figure 6. *Grasp results. The stacked-bar plots show the average distribution of all grasps over all scenes. The stable and slipped grasps are considered successful grasps, where the object is held in the hand after lifting. The gray area shows the proportion of scenes where the methods do not suggest any grasps.*

4.1. Results

The grasp results of the baseline method are shown in Figure 6 and the grasp quality of the successful grasps are in Table 1. The results indicate that the three-finger encompassing-grasps are most successful, followed by the two-finger encompassing-grasps. Due to missing visual information about the back of the objects, the two-finger pinch grasps results more often in collisions or no grasp is suggested. Figure 7 shows a similar general picture, and indicates that all methods

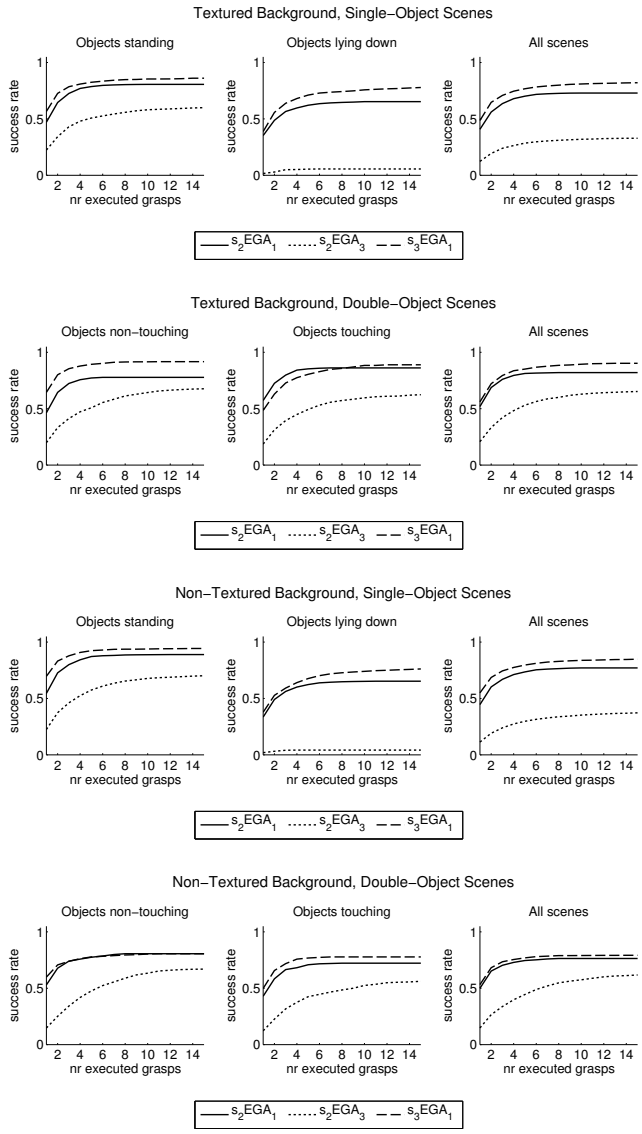

Figure 7. The grasp success rate as a function of the number of attempted grasps.

benefit from successive grasp attempts. For the two and three-finger encompassing grasps, the performance gets to high levels already for a few extra attempts. Table 2 indicates that the two-finger encompassing grasp finds a stable grasp faster than its three-finger counterpart, although it fails to suggest a successful grasp on a larger number of scenes. In general, the methods are more successful in grasping one object from the double-object scene then grasping the object in the single-object scene. However in the double-object scenes there are more collisions. The results for the scenes with textured and non-textured background are very similar, which shows that out method can deal with a higher degree of visual complexity. The grasp success for the individual objects are given in Tables 3 and 4.

Table 1. The lift quality and grasp wrench-space quality for the textured scenes. The values are the averages over the successful trials (stable, slipped).

Textured background

		s_2EGA_1		s_2EGA_3		s_3EGA_1	
		Q_l	Q_{GWS}	Q_l	Q_{GWS}	Q_l	Q_{GWS}
Single	standing	0.58	0.46	0.44	0.30	0.72	0.59
	laying	0.31	0.31	0.04	0.03	0.55	0.50
	all	0.44	0.39	0.24	0.17	0.64	0.55
Double	apart	0.60	0.47	0.46	0.35	0.76	0.63
	close	0.68	0.46	0.45	0.35	0.74	0.60
	all	0.64	0.47	0.45	0.35	0.75	0.62

Non-textured background

		s_2EGA_1		s_2EGA_3		s_3EGA_1	
		Q_l	Q_{GWS}	Q_l	Q_{GWS}	Q_l	Q_{GWS}
Single	standing	0.60	0.46	0.51	0.37	0.79	0.65
	laying	0.36	0.31	0.01	0.03	0.55	0.51
	all	0.48	0.39	0.26	0.20	0.67	0.58
Double	apart	0.62	0.50	0.46	0.40	0.68	0.55
	close	0.51	0.41	0.36	0.35	0.68	0.52
	all	0.56	0.46	0.41	0.38	0.68	0.53

Table 2. The proportion of scenes with a successful (stable or slipped) grasp (p) and the average number of grasp attempts until a successful grasp (ga)

Textured background

		s_2EGA_1		s_2EGA_3		s_3EGA_1	
		p	ga	p	ga	p	ga
Single	standing	0.81	1.77	0.61	3.35	0.88	2.18
	laying	0.65	2.07	0.06	2.38	0.81	3.05
	all	0.73	1.91	0.33	3.27	0.84	2.60
Double	apart	0.78	1.72	0.69	4.12	0.92	1.64
	close	0.86	1.59	0.64	3.79	0.89	2.35
	all	0.82	1.65	0.67	3.96	0.90	1.99

Non-Textured background

		s_2EGA_1		s_2EGA_3		s_3EGA_1	
		p	ga	p	ga	p	ga
Single	standing	0.89	1.80	0.72	4.13	0.94	1.53
	laying	0.65	2.17	0.04	1.78	0.81	5.04
	all	0.77	1.96	0.38	4.00	0.88	3.15
Double	apart	0.81	1.68	0.69	4.75	0.81	1.58
	close	0.72	1.75	0.58	4.94	0.78	1.62
	all	0.76	1.71	0.64	4.84	0.79	1.60

5. Discussion

We presented VisGraB, a database and a software tool for benchmarking vision-based grasping of unknown objects. The database contains real stereo images, which can be used by the user to generate grasp hypotheses. These hypotheses can then be passed on to the software tool, which contains a dynamic grasps simulator that plans, executes, and tests the grasp. The database contains a large set of scenes, with different objects displaying a variety of different shapes, sizes, colors and textures, and with different backgrounds. By performing the grasps in simulation, a large number of grasps can be repeatedly tested. The benchmark facilitates 1) the evaluation and comparison

Table 3. *Percentage of successful grasps for the different objects in the textured scenes. Results for the single-object scenes are split into standing (s) and laying (l) object poses and for the double-object scenes into far (f) and close (c). The pairs in the double-object scenes are: a: 1-18, b: 2-11, c: 3-7, d: 4-15, e: 5-14, f: 6-8, g: 9-13, h: 10-12, i: 16-17.*

Textured background

Single-object scenes

	s_2EGA_1		s_2EGA_3		s_3EGA_1	
	s	l	s	l	s	l
1	50%	10%	12%	8%	76%	31%
2	28%	74%	4%	0%	77%	86%
3	17%	32%	21%	6%	9%	19%
4	48%	15%	38%	0%	37%	21%
5	57%	38%	50%	0%	33%	12%
6	45%	44%	9%	0%	67%	12%
7	23%	76%	43%	0%	35%	64%
8	33%	44%	0%	0%	31%	13%
9	23%	9%	16%	0%	38%	55%
10	87%	29%	31%	4%	97%	61%
11	60%	47%	20%	0%	59%	73%
12	54%	60%	5%	0%	76%	47%
13	54%	38%	22%	0%	59%	58%
14	74%	4%	46%	0%	86%	35%
15	14%	19%	12%	0%	63%	41%
16	90%	50%	64%	0%	82%	65%
17	33%	25%	0%	0%	13%	0%
18	13%	0%	0%	8%	53%	0%

Double-object scenes

	s_2EGA_1		s_2EGA_3		s_3EGA_1	
	f	c	f	c	f	c
a	58%	87%	0%	0%	84%	85%
b	21%	48%	0%	0%	40%	55%
c	0%	32%	36%	33%	45%	31%
d	37%	48%	24%	23%	42%	24%
e	52%	66%	43%	30%	76%	61%
f	44%	54%	20%	25%	43%	39%
g	31%	33%	15%	27%	68%	36%
h	53%	76%	8%	9%	76%	55%
i	58%	50%	10%	8%	67%	31%

Table 4. *Percentage of successful grasps for the different objects in the non-textured scenes. Results for the single-object scenes are split into standing (s) and laying (l) object poses and for the double-object scenes into far (f) and close (c). The pairs in the double-object scenes are: a: 1-18, b: 2-11, c: 3-7, d: 4-15, e: 5-14, f: 6-8, g: 9-13, h: 10-12, i: 16-17.*

Non-textured background

Single-object scenes

	s_2EGA_1		s_2EGA_3		s_3EGA_1	
	s	l	s	l	s	l
1	72%	25%	7%	0%	93%	30%
2	37%	69%	4%	0%	85%	88%
3	34%	51%	23%	0%	49%	10%
4	69%	5%	28%	0%	53%	12%
5	52%	21%	31%	0%	48%	13%
6	43%	19%	8%	0%	66%	6%
7	41%	63%	23%	0%	46%	61%
8	25%	50%	13%	0%	21%	0%
9	48%	33%	10%	0%	85%	61%
10	86%	19%	17%	0%	99%	72%
11	70%	46%	20%	0%	89%	74%
12	33%	54%	6%	0%	74%	47%
13	70%	29%	21%	15%	91%	56%
14	73%	13%	24%	0%	87%	39%
15	45%	14%	10%	0%	86%	19%
16	78%	44%	66%	0%	91%	53%
17	19%	0%	16%	18%	19%	0%
18	13%	10%	50%	0%	57%	20%

Double-object scenes

	s_2EGA_1		s_2EGA_3		s_3EGA_1	
	f	c	f	c	f	c
a	71%	79%	3%	3%	93%	82%
b	42%	24%	2%	0%	44%	53%
c	29%	4%	17%	4%	16%	28%
d	38%	45%	18%	18%	38%	54%
e	66%	63%	27%	21%	83%	48%
f	77%	55%	18%	22%	38%	51%
g	25%	44%	12%	17%	64%	80%
h	57%	20%	7%	3%	63%	19%
i	35%	26%	19%	19%	74%	27%

of different vision-based grasp-generation methods in a standardized fashion, and 2) a focus on the vision methods instead of on the robotic hardware. We presented an example as an illustration of the use of the benchmark.

In addition to what we presented here, the VisGraB framework can be used for evaluating a variety of tasks related to grasping, for example grasping known objects can be tested using the KIT object models, and learning methods can be evaluated on their generalization abilities. Although in VisGraB, we focus on the generation of grasp hypotheses from a pair of stereo images and outsource the grasp execution to the grasp simulator, RobWork can simulate visual and tactile observations, allowing implementations of closed-loop grasp execution.

We strongly encourage the use of the benchmark to test your vision based grasp-generation methods and to compare it to other methods. We are very open to extend the benchmark based on future needs from the community.

Acknowledgments

This work was supported by the EU through the projects eSMCs (FP7-IST-270212), XPERIENCE (FP7-ICT-270273), and CogX (FP7-ICT-215181), by the Swedish Foundation for Strategic Research through the project RoSy.

References

[1] Y. Bekiroglu, J. Laaksonen, J. A. Jørgensen, V. Kyrki, and D. Kragic. Assessing grasp stability based on learning and haptic data. IEEE Transactions on Robotics, 27(3):616-629, 2011

[2] L. Bodenhagen, D. Kraft, M. Popović, E. Baśeski, P. E. Hotz, and N. Krüger. Learning to grasp unknown objects based on 3d edge

information. In Proceedings of the 8th IEEE international conference on Computational intelligence in robotics and automation, 2009
[3] J. Bohg and D. Kragic. Learning grasping points with shape context. Robotics and Autonomous Systems, 58(4):362-377, 2010
[4] J. Cortsen, J. A. Jørgensen, D. Sølvason, and H. G. Petersen. Simulating robot handling of large scale deformable objects: Manufacturing of unique concrete reinforcement structures. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2012
[5] J. Cortsen and H. G. Petersen. Advanced off-line simulation framework with deformation compensation for high speed machining with robot manipulators. IEEE – ASME Transactions on Mechatronics, 2012
[6] N. Curtis and J. Xiao. Efficient and effective grasping of novel objects through learning and adapting a knowledge base. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008
[7] S. El-Khoury and A. Sahbani. Handling objects by their handles. In Proceedings of IROS 2008 Workshop on Grasp and Task Learning by Imitation, 2008
[8] L.-P. Ellekilde and J. A. Jørgensen. Usage and verification of grasp simulation for industrial automation. In Proceedings of Automate 2011, Chicago, Illinois, 2011
[9] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. International Journal of Computer Vision, 88(2):303-338, June 2010
[10] C. Ferrari and J. Canny. Planning optimal grasps. In Proceedings of the 1992 IEEE International Conference on Robotics and Automation (ICRA), Nice, France, 1992
[11] C. Goldfeder, P. K. Allen, C. Lackner, and R. Pelossof. Grasp planning via decomposition trees. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'07), 2007
[12] C. Goldfeder, M. Ciocarlie, H. Dang, and P. K. Allen. The columbia grasp database. In Proceedings of the International Conference on Robotics and Automation (ICRA), Kobe, Japan, 2009
[13] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007
[14] K. Hübner, S. Ruthotto, and D. Kragic. Minimum volume bounding box decomposition for shape approximation in robot grasping. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'08), 1628-1633, 2008
[15] J. A. Jørgensen, L. P. Ellekilde, and H. G. Petersen. RobWorkSim - an open simulator for sensor based grasping. In Proceedings of Joint 41st International Symposium on Robotics (ISR 2010) and the 6th German Conference on Robotics, Munich, 2010
[16] J. A. Jørgensen, A. R. Fugl, and H. G. Petersen. Accelerated hierarchical collision detection for simulation using cuda. In Proceedings of the Seventh Workshop on Virtual Reality Interactions and Physical Simulations (VRIPHYS 2010), 97-104, Copenhagen, Denmark, 2010
[17] J. A. Jørgensen and H. G. Petersen. Usage of simulations to plan stable grasping of unknown objects with a 3-fingered schunk hand. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Nice, France, 2008
[18] J. A. Jørgensen and H. G. Petersen. Grasp synthesis for dextrous hands optimised for tactile manipulation. In Proceedings of the Joint 41st International Symposium on Robotics (ISR 2010), Münich, Germany, 2010
[19] A. Kjær-Nielsen, A. G. Buch, A. E. K. Jensen, B. Møller, D. Kraft, N. Krüger, H. G. Petersen, and L.-P. Ellekilde. Ring on the hook: Placing a ring on a moving and pendulating hook based on visual input. Industrial Robot: An International Journal, 38(3):301-314, 2011
[20] G. Kootstra, M. Popović, J. A. Jørgensen, D. Kragic, H. G. Petersen, and N. Krüger. VisGraB: A benchmark for vision-based grasping. http://visgrab.sdu.dk
[21] G. Kootstra, M. Popović, J. A. Jørgensen, K. Kuklinski, K. Miatliuk, D. Kragic, and N. Krüger. Enabling grasping of unknown objects through a synergistic use of edge and surface information. International Journal of Robotics Research, under review
[22] B. León, S. Ulbrich, R. Diankov, G. Puche, M. Przybylski, A. Morales, T. Asfour, S. Moisio, J. Bohg, J. Kuffner, and R. Dillmann. Opengrasp: a toolkit for robot grasping simulation. In Proceedings of the Second international conference on Simulation, modeling, and programming for autonomous robots, SIMPAR'10, pages 109-120, Berlin, Heidelberg, 2010, Springer-Verlag
[23] A. T. Miller and A. T. Miller. Graspit!: A versatile simulator for robotic grasping. IEEE Robotics and Automation Magazine, 11:110-122, 2004
[24] A. L. Olsen and H. G. Petersen. Inverse kinematics by numerical and analytical cyclic coordinate descent. Robotica, 29(3):619-626, 2011
[25] C. Papazov and D. Burschka. Stochastic optimization for rigid point set registration. In Proceedings of the 5th International Symposium on Visual Computing (ISVC'09), volume 5875 of Lecture Notes in Computer Science, pages 1043-1054. Springer, 2009
[26] R. Pelossof, A. Miller, P. Allen, and T. Jebara. An SVM learning approach to robotic grasping. In Proceedings of the IEEE Conference on Robotics and Automation, 2004
[27] M. Popović, G. Kootstra, J. A. Jørgensen, D. Kragic, and N. Krüger. Grasping unknown objects using an early cognitive vision system for general scene understanding. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2011.
[28] M. Popović, D. Kraft, L. Bodenhagen, E. Başeski, N. Pugeault, D. Kragic, T. Asfour, and N. Krüger. A strategy for grasping unknown objects based on co-planarity and colour information. Robotics and Autonomous Systems, 58(5):551-565, 2010
[29] N. Pugeault, F. Wörgötter, and N. Krüger. Visual primitives: Local, condensed, and semantically rich visual descriptors and their applications in robotics. International Journal of Humanoid Robotics (Special Issue on Cognitive Humanoid Vision), 7(3):379-405, 2010
[30] D. Rao, Q. V. Le, T. Phoka, M. Quigley, A. Sudsang, and A. Y. Ng. Grasping novel objects with depth segmentation. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010
[31] A. Saxena, J. Driemeyer, and A. Y. Ng. Robotic grasping of novel objects using vision. The International Journal of Robotics Research, 27(2):157-173, 2008
[32] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International Journal of Computer Vision, 47(1-3):7-42, 2002
[33] R. Stolkin, A. Greig, and J. Gilby. A calibration system for measuring 3D ground truth for validation and error analysis of robot vision algorithms. Measurement Science and Technology, 17:2721-2730, Oct. 2006
[34] S. Ulbrich, D. Kappler, T. Asfour, N. Vahrenkamp, A. Bierbaum, M. Przybylski, and R. Dillmann. The opengrasp benchmark suite: An environment for the comparative analysis of grasping and dexterous manipulation. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2011