Enabling grasping of unknown objects through a synergistic use of edge and surface information

Gert Kootstra∗1, Mila Popović2, Jimmy Alison Jørgensen3, Kamil Kuklinski4, Konstantin Miatliuk4, Danica Kragic1, and Norbert Krüger2

1The Computer Vision and Active Perception Lab, CSC, Royal Institute of Technology (KTH), Stockholm, Sweden
2The Cognitive Vision Lab, the Mærsk Mc-Kinney Møller Institute, University of Southern Denmark, Odense, Denmark
3The Robotics Lab, The Mærsk Mc-Kinney Møller Institute, University of Southern Denmark, Odense, Denmark
4Automation and Robotics Dept. Białystok University of Technology, Poland

Abstract

Grasping unknown objects based on visual input, where no a-priori knowledge about the objects is used, is a challenging problem. In this paper, we present an Early Cognitive Vision system that builds a hierarchical representation based on edge and texture information which provides a sparse but powerful description of the scene. Based on this representation, we generate contour-based and surface-based grasps. We test our method in two real-world scenarios, as well as on a vision-based grasping benchmark providing a hybrid scenario using real-world stereo images as input and a simulator for extensive and repetitive evaluation of the grasps. The results show that the proposed method is able to generate successful grasps, and in particular that the contour and surface information are complementary for the task of grasping unknown objects. This allows for dealing with rather complex scenes.

Keywords: Vision-based grasping, Grasping unknown objects, Visual scene representation, Dexterous hands

1 Introduction

In this paper, we propose a vision system for general scene understanding allowing for grasp planning and execution. We focus on grasping unknown objects for which a-priori knowledge is not available. In contrast to 2D approaches, which often need simplifying assumptions on the actual action execution, e.g., (Saxena et al., 2008; Chinellato et al., 2005), we make use of 3D information in terms of contour and surface descriptors, allowing for improved grasp planning. In contrast to other 3D approaches that are based on a representation of the scene using different kinds of shape primitives, e.g., (Hübner et al., 2008; Miller et al., 2003), our approach operates on elements of a visually extracted hierarchical representation of the scene, which accurately represents the shape of objects and does not require prior segmentation or manual pre-processing.

One of the problems in grasp planning is the nearly infinite number of possible grasps, which all need to be evaluated to assess their quality. Many current approaches therefore reduce the number of possible grasps by modeling the object shape with a number of shape primitives, such as boxes (Hübner et al., 2008), cylinders, cones, spheres (Miller et al., 2003), or superquadrics (Goldfeder et al., 2007). With the approach we present here, such explicit shape abstractions are not necessary. Our vision system inherently provides a sparse and abstract, but powerful set of 3D features. Making use of our hierarchical representation of the scene, the amount of computed grasps can be controlled by the granularity of the feature descriptors at the different levels of the hierarchy. Moreover, our 3D features are naturally aligned with the shape of the object, resulting in an accurate representation of the object’s geometry, which is not necessary the case when shape primitives are used.

More specifically, we propose and evaluate a method for the bottom-up generation of two- and three-finger grasps based on contour and surface structures. These structures are extracted by means of an extension of the biologically-motivated hierarchical vision system (Pugeault et al., 2010a). This system, in the following called Early Cognitive Vision (ECV) system, makes use of an elaborated mid-level ECV stage in which structurally rich and disambiguated information is provided to higher levels of visual processing (for a detailed discussion see (Krüger et al., 2010)). This system has been applied to the problem of grasping unknown objects based on contour relations (Popović et al., 2010). The ECV system has not only been used for grasping, but also for tasks such as pose estimation (Detry et al., 2009; Kjær-Nielsen et al., 2011), and object learning and recognition (Pugeault et al., 2010a; Başeski et al., 2010).

In this paper, we extend the ECV system, which primarily was dealing with edge-like structures (Pugeault et al., 2010a), by texture information to allow for the association of grasps to surface information in addition to the contour information. A visual hierarchical representation is provided for both modalities, where the perceptual organization is guided by 2D and 3D geometrical and appearance relations between visual entities at the different levels of the hierarchy. In the edge domain, the hierarchy consists of local edge primitives, grouped into contours, which are then matched to other contours belonging to
the same surface, resulting in surface contour pairs. In the texture domain, local textured patches (texlets) are grouped into larger surface elements (surflings), which are further grouped into surfaces. In this paper, we explore the usability of the contour and surface representations for grasping. Using the ECV representations, we define different types of Elementary Grasping Actions (EGAs), see Figure 1. An EGA relates a set of perceptual features to a specific grasping action. The left side of the figure shows the three contour-based EGAs, as presented in (Popović et al., 2010). These are a two-finger encompassing grasp targeted at a matched contour pair (Fig. 1a), and a top pinch grasp (Fig. 1b) and a side pinch grasp (Fig. 1c), both targeted at one of the contours. The right side of the figure shows three surface-based EGAs, including a two-finger encompassing grasp (Fig. 1d), a three-finger encompassing grasp (Fig. 1e), and a two-finger side pinch (Fig. 1f). Surface-based EGAs are defined for both levels of surface representation. In this paper, we investigate the performance of these different EGAs.

To test the performance of the different EGAs, we perform systematic tests of the generated grasp hypotheses in two scenarios. First, by using the vision-based grasping benchmark, VisGraB (Kootstra et al., under review), which provides a mixed real-world and simulated environment, in which features are extracted from real visual data and grasps are performed in a virtual environment using a dynamic simulation. This allows us to test a large number of grasps generated from natural stereo images (in total over 40,000 grasps were tested). By that, we can make elaborated quantifications of contour-based and surface-based grasps. The second scenario is a real-world scenario with a stereo camera, an industrial robot arm, a gripper, and real objects. Grasps are tested in two different hardware setups, one with the Schunk parallel jaw gripper and the other with the Schunk three-finger dexterous hand. Some examples of grasps performed in the different setups can be appreciated in Fig. 1.

This paper makes the following three contributions: (1) We extend the ECV system with a hierarchy of features in the texture domain, providing a sparse and meaningful representation of the scene. The representation on one side reduces the search space for grasping and on the other side creates additional context information which is relevant for grasping. (2) We define new grasping affordances based on surface information on two different levels of abstraction, using an abstract rectangular description as well as using a concrete description of surface-boundary features and build a system that performs the grasps, and (3) we show the complementary strength of edge and texture information for grasping in an extensive experimental evaluation. This paper is a major extension of the conference article (Popović et al., 2011): First, a novel surface-based grasp-generation method is laid out. Second, a much more detailed description of the applied methods is given. Finally, elaborate experiments in a mixed real-world and simulated setup, as well as in two different real robotic setups are presented. The implementation of the methods discussed in this paper can be downloaded from http://www.covig.org/grasping and used under a BSD license.

The paper is organized as follows: Section 2 discusses the related work in the area of vision-based grasping. In Section 3, we introduce the ECV system. Section 4 presents the three grasp generation algorithms and details on the grasp execution procedure are given in Section 5. We describe the experimen-
tal evaluation scenarios and give the overview of the results in Section 6. The paper is concluded with a discussion in Section 8.

2 Related Work

Different approaches to visual-based object grasping have been proposed by the robotic community. As proposed in (Bohg and Kragic, 2010), these approaches can be roughly divided into grasping of known, familiar, and unknown objects.

For grasping known objects, a detailed 2D or 3D model of the object is generally available. This model is then fitted to the current visual observation to retrieve the pose of the object. Based on the model and the pose estimation, a large number of grasps suggestions are generated and their quality is evaluated to select the most promising grasp, e.g., (Nguyen, 1989; Shimoga, 1996). One of the main challenges is the huge amount of possible grasps. In order to reduce the search space, different techniques have been applied. In (Miller et al., 2003; Goldfeder et al., 2007; Hübner et al., 2008), the shape of the object is simplified by using shape primitives, such as, spheres, boxes, and superquadrics, thereby reducing the number of possible grasps. Another method for reducing the dimensionality of the configuration space of the hand, using so-called eigen-grasps, has been proposed in (Ciocarlie and Allen, 2009). It has been demonstrated in (Borst et al., 2003) that generating an optimal grasp, according to some quality measure, is not necessary. A small random subsample of the possible grasps will generate grasps of the average quality which is sufficient. In (Detry et al., 2011) grasp affordances are modeled with continuous probability density functions (grasp densities) which link object-relative grasp poses to their success probability. Grasp data is gathered from exploration and when a satisfactory number of data is available, an importance-sampling algorithm turns these into a grasp density.

The above-mentioned studies have been done in simulation, assuming complete knowledge about the object and the robot, and ignoring noise, with the exception of (Detry et al., 2011) and (Hübner et al., 2008), where incomplete and noisy data has been used as well. Furthermore, the studies all assume a perfect segmentation of the object from its background. In contrast, we propose a method based on real visual data of cluttered scenes without any knowledge about the presented objects.

In work on the grasping of familiar objects, the system is generally trained on a set of objects and learns the relation between some visual features and the grasp quality. This knowledge is then used to grasp resembling objects. In (El-Khoury and Sahbani, 2008), for instance, the grasplability of object parts is learned based on the parameters of superquadrics fitted to segmented parts of the object and using human expertise. An
SVM has been trained to predict the grasp quality based on the hand configuration and the parameters of a single-superquadric representation of the objects in (Pelossof et al., 2004). In (Curtis and Xiao, 2008), grasp knowledge is learned on a set of simple geometrical shapes and applied to grasp novel objects. All these experiments were done completely in simulation with synthesized data. In (Saxena et al., 2008) grasping hypotheses are learned based on a set of local 2D images features using synthesized objects, and this knowledge is used to grasp objects in the real world. The feature vector used is a high dimensional set of edge, texture and color features on different scales. Different features of two contours resulting from our ECV system have been used in (Bodenhagen et al., 2009) to learn to predict the grasping success.

When grasping unknown objects, no model of the objects or prior grasp knowledge is used and all reasoning about grasping is done on the visual observation of the scene. In (Hübner et al., 2008) and (Dune et al., 2008), shape primitives, respectively boxes and quadratics, were used to deal with the noisy and incomplete data coming from robotic sensors, and to provide a reduced set of potential grasps. In (Gallardo and Kyrki, 2011), the stereo-vision data was approximated with box and cylinder primitives, which describe an objects overall shape, as well as its location, orientation, and size. Grasps based on the 2D silhouettes of the objects were tested in (Morales, 2004). A learning framework for discovering the visual features that predict a reliable grasp was presented. In (Richstfeld and Vincze, 2008), a top surface of an object was detected by selecting the points that belong to the top 3 mm of the segmented point cloud. The grasping points were placed at the rim of the detected surface and the grasps were performed in simulation. A sophisticated 3D representation of the scene based on ECV system was used in (Kraft et al., 2008; Popović et al., 2010) for grasp planning. We build upon this work by extending the system to not only take edge features into consideration, but also texture features.

Most of the studies on vision-based grasping assume a segmentation of the objects from their background. However, for grasping unknown objects in real-world situations this assumption does not hold. When using pinch grasps, (as done in e.g., (Saxena et al., 2008)), single image points are sufficient to define a grasp, making object segmentation to relate multiple points on the same object unnecessary. Input of the user is taken to initialize object segmentation using active contours in (Dune et al., 2008). In (Rao et al., 2010), a bottom(255,909),(744,939)(255,909),(744,939)-up segmentation method based on color and depth information is used and the graspability of the segments is learned using an SVM. In (Popović et al., 2010), we associated two grasp points with the same surface of an object by using co-planarity and co-colority.

In this paper, we present vision-based bottom-up methods for grasping unknown objects, based on unsegmented real-world scenes. Unlike other approached discussed in this section, we do not use a simplification of the object(s) using shape primitives to abstract the shape. Instead we extend the ECV system to produce a sparse, yet semantically meaningful representation of the scene that remains close to the true shapes of the objects and which allows the system to utilize the potential of edge as well as texture information.

3 The Early Cognitive Vision System

The framework of the Early Cognitive Vision system provides a rich hierarchical visual representation that includes edges and textured surfaces (Krüger et al., 2010; Puegault et al., 2010b; Popović, 2012). Fig. 2 shows different stages in creating the hierarchical representation. The left-hand branch illustrates the propagation of edge information, and the right-hand branch shows the propagation of texture information.

The representation is layered and starts with extracting sparse local features from 2D images. These basic features are called multi-modal primitives and encode both geometric and appearance information. In this paper, we use edge primitives and textured-surface primitives (texlets). By matching 2D features over two stereo views, the system derives corresponding 3D descriptors for the different structures, see Fig. 2.

On the second level, the ECV system organizes basic features into perceptual groups (in both 2D and 3D) (Bašek et al., 2010; Puegault et al., 2010b). Edge primitives are grouped into contours, while the texlets are organized in so-called surflings, see Fig. 2c-iii and s-iii. On this higher level of abstraction it is again possible to group the features or to observe their relations. Contours are matched to other contours that are part of the same surface and surflings are combined into surfaces.

The hierarchical organization of our system creates a robust representation of the scene. By sequentially grouping elements into higher-level features using different Gestalt principles (El lis, 1938; Koffka, 1935) (such as proximity, co-linearity, co-planarity, co-colority, parallelism, and symmetry) noise is effectively filtered out. Based on these principles, structure in the data is required, which prevents spurious features to disturb the representation at higher levels.

The contour extraction is described briefly in Sect. 3.1 and we refer to our previous work for more details. The surface extraction is described in a more detail in Sect. 3.2, as it is one of the contributions of this paper, and it allows for the precise formalization of the novel surface-based grasp methods.

3.1 Edges and Contours

Line segments in the ECV system are local edge-feature descriptors (see Fig. 2c-i,c-ii) that integrate geometrical (position and orientation) and appearance (color and phase) information, see (Puegault et al., 2010a). The local edge features are grouped into bigger perceptual groups – contours – based on multi-modal constraints including proximity, col-linearity, co-circularity, and similarity in appearance (see Fig. 2c-iii). Contours, as features on the higher level of abstraction, can again be compared and grouped by observing relations between them. As explained in (Popović et al., 2010), we use a number of Gestalt principles to identify pairs of contours belonging to the same surface of an object. First, we require that both contours are in the same plane (co-planarity). Second, the colors at either side of one contour need to be symmetrical to

---

1http://www.covig.org/
the colors flanking the other contour (symmetry). Finally, the contours are required to be parallel to each other (parallelism). Using these principles, we achieve a robust set of matching contour pairs. These pairs are used to trigger contour-based grasping actions as further explained in Sect. 4.1. For details on the contour extraction, we refer the reader to (Bašeski et al., 2010).

3.2 Texlets, Surflings and Surfaces

In this section, we introduce the surface-domain of the ECV system, including the formalization of texlets, surflings, and surfaces relevant for defining the elementary grasping actions. Some more details can be found in the Ph.D. dissertation of Popović (Popović, 2012).

Texlets: Texlets are visual features that describe the local properties of a textured surface patch. A texlet stores the mean color, the 3D position and the 3D normal of the patch. A texlet is defined as:

$$\Pi^T = (p^T, n^T, c^T)$$

(1)

where $p^T$ is the position of the center of the patch in 3D Cartesian space, $n^T$ is the 3D normal of the surface patch, and $c^T$ is the average color in the CIeLAB color space.

Initial texlets in 2D are extracted from the left image of the stereo pair (see Fig. 2s-i). The image plane is separated into local patches by applying a hexagonal grid. For each cells that contain texture information, a 2D texlet is defined and its color is determined by averaging over the pixels in the cell. Using dense stereo matching on the underlying pixel level, the 3D position and orientation are determined to obtain the 3D texlet (see Fig. 2s-ii).

Due to the hexagonal-grid, each 2D texlet has up to six neighboring texlets. After constructing the 3D texlets, this neighboring structure is propagated from 2D to 3D. The original connection between two neighboring texlets remains if it satisfies the Gestalt principles of proximity and co-colority, meaning that the texlets should be close in 3D position and orientation, and have similar colors. The remaining connections are propagated using the transitivity relation to derive pools of connected texlets sharing similar properties.

However, due to noise, it is possible that wrong connections are created. These local erroneous connections might lead to errors in the segmentation on the global level. To prevent this, we use multiple levels of granularity in the hierarchy, so that local errors do not propagate. Therefore, instead of grouping texlets into surface directly, we utilize surflings as an intermediate level.

Surflings: In the next level of the hierarchy, the texlets are grouped into semi-global surface descriptors called surflings, see Fig. 2s-iii. The system subdivides the pools of texlets into small subsets of about five to ten texlets using k-means clustering on the position. These subsets of similar texlets form the surfling features, which represent the mid-level abstraction bridging the texlets and the surfaces. A surfling is defined as:

$$\Psi^SL = (p^SL, o^SL, s^SL, b^SL, n_b^SL, p_b^SL)$$

(2)

where the surfling feature is a rectangular planar patch described by a full 6D pose (3D position $p^SL$, 3D orientation $o^SL$), size $s^SL$ (width, $l_x^SL$, and length, $l_y^SL$), and color $c^SL$, see Fig. 2s-iii. The boundary label $b^SL \in \{\text{true, false}\}$ tells if the surfling is located at the boundary of the segmented surface. If so, the boundary normal, $n_b^SL$, is a vector assigned to each boundary surfling, which gives the direction of the local boundary, and $p_b^SL$ is the boundary edge point. The boundary label, boundary normal, and boundary edge-point properties will be explained at the end of this section.

The position of the surfling is the center of mass of the member texlets positions, $p^SL = \frac{1}{N} \sum_{i=1}^{N} p^T_i$. The orientation is obtained by fitting a plane through the positions of the member texlets using singular value decomposition (SVD). Let $x^SL$ and $y^SL$ be the eigenvectors with the largest and second largest eigenvalue, then the orientation of the surfling is defined as the rotation matrix $o^SL = (x^SL, y^SL, z^SL)$, where $z^SL = x^SL \times y^SL$. The surface of the surfling lies in the $(x^SL, y^SL)$-plane, while $z^SL$ defines the surface normal. The width, $l_x^SL$, and length, $l_y^SL$, of the surfling are calculated based on the eigenvalues resulting from the SVD. The size in the direction $i$ is given as $l_i^SL = 4 \cdot \sqrt{E_i}, i \in \{x^SL, y^SL\}$, where $E_i$ is the eigenvalue of the corresponding eigenvector. The color property of the surfling is defined in the CIeLAB color space and is derived by averaging the color of the underlying texlets: $c^SL = \frac{1}{N} \sum_{i=1}^{N} c^T_i$.

Surfaces: The ECV system constructs surfaces by grouping surflings in the next level of the hierarchy. The surflings are grouped based on the Gestalt principles of proximity and co-planarity, meaning that surflings that are close in 3D space and lie approximately in the same plane are clustered. Color information is not considered in this grouping step, allowing for heterogeneously colored surfaces to be segmented as one. Color is often an important cue to find surface boundaries and has therefore been used at the lower level to group texlets of similar color into surflings.

To obtain the surfaces, the system establishes a neighboring structure between surflings with proximate position and similar orientation. The surflings are then clustered using the transitivity relation, similarly to how texlets are grouped into surflings. A surface, $\Psi^S_i$, is defined by the set of $M$ surflings that it contains:

$$\Psi^S_i = \{\Psi^SL_1, \ldots, \Psi^SL_M\}$$

(3)

Once the surfaces are computed, the system identifies a subset of member surflings that are positioned on the boundaries of the surfaces. Such a boundary surfling, $\Psi^BL_p$, is labeled with $b^SL = \text{true}$ and has the additional boundary normal property, $n_b^SL$. The boundary normals lie within the surface plane of the surfling and point out of the surface as indicated by the red lines in Fig. 3a-i and 3a-ii.

The boundary normal of a boundary surfling is determined by the two vectors connecting the center of the surfling $p^SL_p$ with the centers of its two closest boundary surflings $p^SL_{b1}$ and $p^SL_{b2}$. These vectors $v_1 = p^SL_{b1} - p^SL_p$ and $v_2 = p^SL_{b2} - p^SL_p$ are
4 Grasp-Generation Methods

The hierarchical visual representation created by the ECV system as defined in Sect. 3 provides an abstract description of the scene in terms of two hierarchies corresponding to contours and surface information. In this section, we apply this general scene representation to the problem of grasping unknown objects. By matching contours and finding surfaces of objects, we can discard many inadequate grasps and thereby drastically reduce the search space compared to creating grasps based on more local descriptors on a lower level of the hierarchy.

In this paper, we look at three methods for generating Elementary Grasping Actions (EGAs), where an EGA links visual features, i.e., contour or surface features, to a specific grasping action. The first method generates two-finger grasps based on contour features, c₁EGAs, see Fig. 1a-c. This method was already proposed in (Popović et al., 2010) and is briefly described in Sect. 4.1. The other two methods use surface features to generate two-finger and three-finger grasps, s₂EGAs, s₃EGAs, see Fig. 1e-f. The first surface-based method approximates the shape of the surface in a coarse way. This method is described in Sect. 4.2.1. The second surface-based method is based on the boundary surfings of the surface and is described in detail in Sect. 4.2.2. Sect. 4.3, finally, describes the filtering of features based on the detection of the supporting plane.

Throughout the paper, we use the following notation for the grasp types: \( [c_s] \)-EGAs, where \( c \) denotes a contour-based grasp and \( s \) a surface-based grasp. \( f \) is the number of fingers involved in the grasp, and \( i \) is the index of the grasp subtype, where \( i = 1 \) for encompassing grasps, \( i = 2 \) for top pinch grasps, and \( i = 3 \) for side pinch grasp. As an example, \( s₂EGA \) is a two-finger surface-based encompassing grasp, see Fig. 1.

We choose to define a general grasp \( G \) as:

\[
G = (\mathcal{C}, \mathbf{a}_G)
\]  

where \( \mathcal{C} \) is a set of two, \( \mathcal{C} = \{c_1, c_2\} \), or three, \( \mathcal{C} = \{c_1, c_2, c_3\} \) contacts, and \( \mathbf{a}_G \) is the approach direction of the gripper, see Fig. 3b. A contact \( c \) is defined by its position, \( \mathbf{c}_{pos} \), and normal, \( \mathbf{c}_n \), see Fig. 3a-iv:

\[
\mathbf{c} = (\mathbf{c}_{pos}, \mathbf{c}_n)
\]  

A contact defines the place and orientation that is the target for one of the gripper’s fingers. For the encompassing grasps, the contacts mark the exact location of the predicted meeting between the fingers and the object. For the pinch grasps, however, we do not know the thickness of an edge or a surface, since we do not reason about this. To deal with this, for the top pinch grasps, the contact of one finger is therefore placed in the middle of the presumed opening and the other at the same distance outside of the object. For the side pinch grasp, the contacts are placed so that the gripper’s fingers are at a fixed distance from the feature. A distance of 20 mm has been used for the experiments in this article.

The following subsections describe the computation procedures for the three grasping methods in detail.
The approach direction of the gripper is the inverse of the common plane normal. As the placement of the opening below the surface is not known, the exact placement of the fingers is set to a fixed predefined value. The approach direction of the gripper is orthogonal both to the local contour direction and to the common plane normal.

\[ \text{c}_2 \text{EGA}_3 \] is a side pinch grasp that aims at grasping the surface represented by the common plane. It assumes that there is an opening just below the surface, see Fig. 4d. The contacts are positioned at either side of the contour center, along the direction of the common plane normal. As the placement of the opening below the surface is not known, the exact placement of the fingers is set to a fixed predefined value. The approach direction of the gripper is orthogonal both to the local contour direction and to the common plane normal.

For more details on the contour elementary grasping actions, we refer to (Popović et al., 2010).

4.2 Surface Elementary Grasping Actions – \( s \text{EGAs} \)

The textural hierarchy presented in Sect. 3.2 identifies surfaces of the objects in the scene. This information is used for two- and three-finger Surface Elementary Grasping Actions, with \( s_2 \text{EGA}_1 \) (Fig. 1d) and \( s_3 \text{EGA}_1 \) (Fig. 1e) being encompassing grasps and \( s_2 \text{EGA}_3 \) (Fig. 1f) being a pinch grasp. In this work, we investigate two methods for generating grasps based on surface features, both using a different level of abstraction. The first method, discussed in Sect. 4.2.1, relies on modeling an approximate shape of the surface in a coarse way. It produces few two-finger grasps that capture the surface as a whole. The second method explained in Sect. 4.2.2, is more elaborate. It operates on individual boundary surflings, which allows for a larger number of grasps and moreover for grasps that involve three fingers.

4.2.1 The SVD Method – \( s \text{EGAs} \)

For each extracted surface, \( \Psi^s \), we perform a singular value decomposition (SVD) on the 3D positions of the boundary sur-
flings, $p_i^{SL}$. This provides the center of mass and the eigenvectors, which we use to construct a local reference frame $\mathcal{L}$. The center of mass, $M$, is used as the origin of $\mathcal{L}$ and the eigenvectors ordered by descending eigenvalues, $x, y, z$, form the $x$-, $y$-, and $z$-axis of $\mathcal{L}$, see Fig. 5b. The $x$-axis and $y$-axis also define a plane $P$ fitted through the positions of the boundary surflings, while the $z$-axis defines the normal $p = z$ of the plane $P$.

In order to position the contacts, we need to estimate the dimensions of the area covered by the surface. We do so by modeling the surface as a minimum rectangle covering the boundary surflings projected to the plane $P$. The position of the projected boundary surflings is indicated as $p_i^{SL}$. The rectangle is lying in the $xy$-plane and its orientation is aligned with the $x$- and $y$-axis. See Fig. 5c for an illustration of the estimated rectangle.

For each surface, we produce two encompassing and four pinch grasps, $G = \{c_1, c_2\} \cup \{a_d\}$, i.e. in total six grasps of the surface along the main directions, see Figs. 5d and 6. For encompassing ($s_2\text{EGA}_3$) grasps, the contact positions $p_{pos}$ are located at the centers of two opposing edges of the estimated rectangle. The contact normals $c_n$ are parallel to the $x$ or $y$-axis and are pointing towards the center of the rectangle. The approach direction of the gripper is the inverse of the surface normal: $a_d = -p_n$. For pinch grasps ($s_2\text{EGA}_3$), the contacts are surrounding the centers of the boundaries such that $c_1$ and $c_2$ are exactly above and below the centers of the edges (in the local coordinate frame). The height above or below the $xy$-plane is set to a predefined value depending on the scale of the gripper. The contact normals $c_n$ are pointing vertically towards the $xy$-plane, aligned with $p_n$. The approach directions of the gripper are parallel with either the $x$ or $y$-axis, see Fig. 5d.

### 4.2.2 The Boundary Method – $s_2\text{EGA}_3$

The ECV system provides information about the locations of the boundaries of the surfaces. A boundary is represented in a sparse way, through properties of boundary surflings $\Psi_b^{SL}$, see Sect. 3.2 and Fig. 3. In this method we fit a plane $P$ through the 3D positions of boundary surflings, with the plane normal $p_n$ in the same way as described in the Sect. 4.2.1. We then project the boundary surflings to this plane. We notate the projected values with a ‘$p$’ in a preceding superscript, i.e., projected position $p^{SL}_1$, projected size $s^{SL}_1$, projected boundary normal $n^{SL}_1$, projected boundary edge point $p^{SL}_E$.

By projecting surflings to a plane, we touch upon the research area of vision-based planar grasps, see e.g. (Morales et al., 2006; Chinellato et al., 2005), where objects are represented by their contours in 2D. The contours are usually derived from a single top-down view of the object and are analyzed to estimate optimal top down grasps. Morales et al. (2006) suggest numerous criteria for choosing the optimal contacts of fingers with the selected boundary regions. These criteria make use of parameters such as contour curvature, distribution of forces and torques, and the deviations of predicted forces and torques due to the kinematics of the robotic hands.

It is important to notice that although we make use of the principles from the planar grasp methods, we overcome some of the weaknesses connected to those methods. The purely 2D method assume the top-down view of the object, and the analyzed 2D plane is always parallel to the horizontal support surface. In our method the graspable plane originates from any surface of the object and can be in an arbitrary orientation. The second weakness of the purely planar methods is that the contour acquired from the top-down view does not inform about the height nor about the 3D shape of the object, it does not necessarily represent an existing surface of the object. In contrast, our method is based on the 3D features of the ECV representation and the grasps are aimed at the specific segmented 3D surfaces.

We now give the definition of $s_2\text{EGA}_3$ pinch grasps, followed by the definition of $s_2\text{EGA}_1$ and $s_3\text{EGA}_3$ encompassing grasps.

#### $s_2\text{EGA}_3$ Pinch Grasps

A pinch surface grasp, $s_2\text{EGA}_3 = \{c_1, c_2\} \cup \{a_d\}$, is defined for each non-corner boundary surfling $\Psi_b^{SL}$, (see Sect. 3.2). A
The selection criteria are based on gripper-specific kinematic constraints and on geometric constraints. We do not apply the commonly used grasp-stability measures based on the wrench space (Miller and Allen, 2004) because these measures usually assume perfect knowledge of the object’s shape. Since we are dealing with visual observations of unknown objects in the real world, the derived shape representation will be somewhat noisy and uncertain. We therefore apply less detailed heuristics instead, with the purpose of maximizing the grasp stability.

We apply the constraints in the order of increased computational complexity. The first constraint is the gripper-specific constraint that filters out the contact combinations that are too far apart, having in mind the maximal distance between the gripper fingers.

We apply two additional geometric constraints both for two finger and three finger grasps. The geometric constraints aim to prevent sliding and to minimize the torques. As the first geometric constraint, we use the Coulomb’s friction model $F_t \leq \mu \cdot F_n$ to derive a friction cone for each contact. The friction coefficient $\mu$ defines a friction half angle: $\beta = \arctan(\mu)$, that is the maximum distance in angle between the contact normal and the direction of the force applied by the finger, at which sliding will not occur, see Fig. 8a. Since the friction coefficient is not known, we empirically set it to a conservative value of 0.3.

In the case of contact pairs, $s_2 \text{EGA}_1$, we first require that the angle between the contact normals is within $n_1 \cdot (-n_2) < \beta$, see Fig. 8b. Additionally we use a contact region $c_R$, which is the region of the boundary that supports the given contact point, see Fig. 3a-iv. The contact region is lying in the plane of the surfacing, goes through the contact point $e_{\text{pos}}$, and is perpendicular to the contact normal $n_R$. The length of the region is given by the size, $s_{\text{SL}}$, of the surfacing. For this stability criterion, we create two areas by projecting the contact regions in the direction of the contact normals, as illustrated in Fig. 8c. We then require that at least one of these areas intersects with the opposite contact region.

For triplets of contacts, $s_3 \text{EGA}_1$, we first require that the contact normals positively span the plane. This is the case if each of the three vectors can be written as a linear combination of the other two using only positive weights. Second, we require that the intersection of friction cones is not empty (Ponce and Faverjon, 1995), see Fig. 8d.

4.3 Filtering of Features based on Supporting Plane Detection

The ECV scene representation contains visual features of all elements in the scene. Since our grasping methods reason on the surface level, they will attempt to grasp every surface in the scene, including the table plane, objects below the table plane, and objects out of reach. To prevent this, we assume that the objects of interest are placed on the table plane and we filter out all other features. Note that without this filter step, the system also makes a sensible representation of the scene. The only
In our algorithm, a number of parameters are used for the grasp collisions, as is described in Sect. 5.2. Over, together with the estimated plane, they are used to predict the estimated plane in green with the remaining features. Only above the plane in their original color. The right column shows the features below the plane in red, and the remaining features that belong to the detected supporting plane in blue, all middle column illustrates the plane detection, with all the features that are above the table plane.

The first column shows the original left camera image. The middle column shows the features belonging to the table plane in blue, those below the plane in red, and the features above the table plane are given in their original color. The right column shows the remaining features after filtering and the estimated plane model.

The reason to apply the filter is to restrict the graspable surfaces to those that are above the table plane. To detect the table plane, we assume that it is identifiable as the dominant horizontal plane in the scene, similar to e.g., (Jain and Kemp, 2010; Björkman and Kragic, 2010). We detect this plane using Random Sample Consensus (RANSAC) (Fischler and Bolles, 1981). Iteratively, three texlets are randomly selected, the plane through those points is calculated, and if this plane is more horizontal than vertical, the texlets that fit the plane both in position (< 10 mm) and orientation (< 30°) are classified as inliers. The iteration with the largest consensus, i.e., largest number of inliers, is taken and the table-plane estimation is refined by fitting a plane through all inliers using singular value decomposition.

All features above the table plane are kept and the rest is filtered out. Examples of this filtering step are given in Fig. 9. The first column shows the original left camera image. The middle column illustrates the plane detection, with all the features that belong to the detected supporting plane in blue, all features below the plane in red, and the remaining features above the plane in their original color. The last column shows the estimated plane in green with the remaining features. Only the remaining features are used to determine the EGAs. Moreover, together with the estimated plane, they are used to predict collisions, as is described in Sect. 5.2.

In our algorithm, a number of parameters are used for the grasp generation:

- **Maximum distance between contact points.** Used for all grasps. This distance depends on the gripper.
- **Distance of contact points from surface for side-pinch grasp.** Larger values will result in less collisions with the target object, but at the risk of getting more collisions with surrounding objects.
- **Thresholds on parallel and opposite contours.** Thresholds to select stable contour-based grasps.
- **Friction coefficient $\mu$.** Used for the selection of stable surface-boundary grasps. Larger values permit more grasps, but lower values make the selection more susceptible to noise.
- **Thresholds on plane fit distance and orientation.** Thresholds to find the dominant plane.

The results are quite robust to the exact parameter values. All parameters have a clear geometrical or physical meaning.

## 5 Grasps Execution

This section describes how the grasps are executed. We first describe how the robot hand configurations are derived from the grasp definitions in Sect. 5.1. Section 5.2 explains how collision detection is used to prevent grasp configurations that are predicted to collide with objects in the scene. The execution of the grasps is described in Sect. 5.3. Finally, we explain the evaluation of the grasps in Sect. 5.4.

### 5.1 Inverse Kinematics

Once a general grasp has been defined through the set of contacts and the approach direction (see, (4)), the inverse kinematics is used to search for the feasible gripper configurations that will place the gripper fingers at or close to the desired locations. The inverse kinematics makes the following mapping:

$$ G = (\{c_1, \ldots, c_n\}, a_d) \mapsto (X_{\text{hand}}, q) \tag{6} $$

where $X_{\text{hand}}$ is the 6-dimensional Cartesian pose of the hand base, and $q$ is the joint configuration. For the parallel jaw gripper, $q$ is a one dimensional vector marking the distance between the fingers. The Schunk dexterous hand (see Fig. 13 has seven degrees of freedom, and $q = \{q_0, \ldots, q_6\}$ therefore gives the angles of the seven joints.

The grasps are defined by the contact positions at the detected features (contours or surface boundaries), which define the extremities of an object. In order to generate stable grasps, the end effectors of the gripper need to go beyond those points and grasp with a certain depth in the approach direction to have the fingers get a better grip on the object. We do this by translating the hand pose in the direction of the approach vector:

$$ X'_{\text{hand}} = X_{\text{hand}} + d \cdot a_d \tag{7} $$

where $d$ is the depth of a grasp. In our experiments, we use $d = 20 \text{ mm}$ both in simulation and in real experiments.

For evaluating grasps in the real setups, we need to compute the inverse kinematics of the robot arm. i.e. find the configuration of the robot arm $Q_{\text{arm}}$ that will place the robot hand in the desired $X'_{\text{hand}}$. We also make sure that the final hand configuration is accessed from the approach direction, and thus define an additional hand configuration $X_{\text{hand}}^a$ that is set as a waypoint before moving to $X'_{\text{hand}}$:

$$ X_{\text{hand}}^a = X_{\text{hand}} - d^a \cdot a_d, \tag{8} $$

and search for the corresponding $Q_{\text{arm}}^a$. The grasps for which either of the $\{Q_{\text{arm}}, Q_{\text{arm}}^a\}$ can not be found are discarded. In our experiments, we use $d^a = 30 \text{ mm}$ both in simulation and in real experiments.
5.2 Collision Detection

Our methods do not consider all global information, but rather generate grasps based on surface information. This potentially can result in grasp hypotheses where the robot would be penetrating other parts of the object, or other objects in the scene. Look for instance at Fig. 6, where some of the suggested grasps will result in the hand penetrating the object. Since such grasp hypotheses are not valid, we aim to prevent this by predicting collisions of the robot and elements in the scene.

The collisions are predicted using a geometrical model of the robotic gripper and a geometrical scene representation. The gripper is modeled using knowledge of its shape and the particular pose and configuration suggested by the grasp hypothesis. The scene is modeled using the detected table plane and the ECV features above the plane, as described in Sect. 4.3 and shown in Fig. 9. When the model of the gripper penetrates any of the geometrical shapes in the scene, the particular grasp hypothesis is discarded.

5.3 Grasp Execution

In order to execute a grasp, the system has to perform preparatory movements. The gripper is first set to a pre-grasp configuration \( q_{\text{open}} \), where the gripper is opened more than \( q \), and the base of the hand is positioned at the approach pose \( X_{\text{hand}}^a \), which is a distance \( d^a \) away from the target hand pose \( X_{\text{hand}} \). Next, the hand is moved to the target pose \( X_{\text{hand}} \) and the fingers begin to close. In the closing of the fingers, the grasp-control policy guides the joint configuration from \( q_{\text{open}} \) to \( q_{\text{closed}} \), where the gripper is closed more than \( q \). The grasping action is finished either when the joints settle in a static configuration, or when the joint configuration \( q_{\text{closed}} \) is reached.

For the parallel jaw gripper, \( q_{\text{open}} = q + a \), and \( q_{\text{closed}} = 0 \), that is, in the opening configuration, the fingers are \( a = 10 \) mm wider. For the Schunk dexterous hand, the joints at the base of the fingers are changed in the opening and closing configurations: \( q_{\text{open}} = q + [\alpha, 0, 0, \alpha, 0, \alpha, 0] \) and \( q_{\text{closed}} = q + [\beta, 0, 0, \beta, 0, \beta, 0] \). In our experiments, we used \( \alpha = -0.5 \) rad in case of one object, \( \alpha = -0.3 \) rad for multiple objects, and \( \beta = 0.2 \) rad. Note that in case of the top pinch grasps, \( c^2 \text{EGA}_3 \), \( a = 0 \) and \( \alpha = 0 \), because one of the fingers is targeted right in between the two contours.

The inverse kinematics and grasp execution have the following parameters:

- Depth of the grasp, \( d \). Depends on the size of the finger tips of the gripper.
- Approach distance, \( d^a \). Determines the distance from the object from where it starts to approach the object and execute the grasp.

5.4 Grasp Evaluation

The object is grasped when, after the grasp action is finished, all fingers are in contact with the object. However, that does not mean that the grasp is stable. In order to test the grasp quality, we perform a dynamic evaluation of the grasp by lifting the object and observing the outcome. The grasps are then evaluated and classified depending on the outcome of the grasping action and the lifting action. In the hybrid real-world and simulated experimental setup, the grasps are executed in simulation, and the outcomes can therefore be precisely registered by the system. In the real-world setups, the experimenter observes the outcome visually. Details on the evaluation procedure are given in the next section.

6 Experiments

To gain insights in the performance of the different grasp-generation methods, they have been tested in two different experimental setups. First, experiments have been performed in a mixed real-world and simulated setup, (in following also called the hybrid setup), using the VisGrab benchmark (Koestra et al., under review), see Sect.6.1. The benchmark provides stereo images of a large set of real scenes, and the grasps are performed using a dynamics based grasp simulator. This enables extensive testing on real visual input and allows for a comparison between methods under the exact same circumstances. Second, experiments have been performed in a real-world setup to demonstrate the performance of the methods running on real robotic systems. Here two scenarios have been used, one using a parallel jaw gripper described in Sect.6.2, and one using a three-finger dexterous hand described in Sect.6.3.

![Image](https://via.placeholder.com/150)

Figure 10: The mixed real-world and simulated experimental setup.
6.1 Hybrid Real-World and Simulated Setup

The methods have been tested on the VisGraB benchmark\(^2\), which we have presented in (Kootstra et al., under review). The benchmark provides a hybrid real-world and simulated experimental setup. Real camera images provide the input to the ECV system and the grasp-generation methods. The produced grasp hypotheses are then tested in a dynamic simulated environment\(^2\). Figure 10 gives an illustration of the hybrid real-world and simulated setup. This setup gives us the possibility to run a large number of trials and to repeat the experiments in the exact same conditions, allowing for a fair comparison among methods, while having to deal with the noise and uncertainty of the real world. A total of 47,269 grasps have been performed in our experiment.

Real stereo-camera images of a large number of grasp scenes are provided by the benchmark. The 18 different objects used are shown in Fig. 11. The objects have various shapes, sizes, colors, and textures. The benchmark contains scenes with one object and scenes with two objects. The single-object scenes contain the 18 objects in two pose conditions, the one containing four scenes where the object stands upright, and the other with four scenes where the object lies down. In the two-object scenes, 9 combinations of objects are included, again with two pose conditions, one with four scenes with the objects placed apart, and the other with four scenes with the objects touching each other. All scenes are recorded in two texture conditions, placed on a non-textured and on a cluttered/textured table. This gives in total \(2 \times (18 \times 8 + 9 \times 8) = 432\) scenes. Some example scenes are given in Fig. 12.

The VisGraB benchmark furthermore contains simulated versions of all the real scenes in order to test the grasp performance in simulation. The objects and models that are used are part of the KIT Object-Models Web Database\(^4\). These models have been obtained using a laser scanner and therefore provide a realistic representation of the scene. Besides the objects, also the table has been placed in the simulated scene. Figure 12 (bottom row) gives some examples of simulated scenes.

The grasps are evaluated in simulation using RobWork\(^5\). RobWork is a simulator for robotic grasping with dynamic capabilities, which has been used in several related experiments (Jørgensen et al., 2010). The simulator has been shown to provide realistic results in (Ellekilde and Jørgensen, 2011), where several thousands of grasps with a parallel gripper in a real robotic setup have been compared to the simulation. In the experiments, we use a simulation of the Schunk dexterous hand, which allows for both two-finger parallel grasping and flexible three-finger grasping, see Fig. 13.

Using a dynamic simulator 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 grasping and lifting the object. In an experimental trial, the quality of the generated grasp is tested as follows: the hand is placed in the determined pose \(X_{\text{hand}}\), and the fingers are opened in the opening configuration, \(q_{\text{open}}\). The fingers then close to the closing configuration, \(q_{\text{close}}\). The object is potentially grasped when the hand settles in a static 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. The result is classified as follows:

**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.

\(^2\)http://www.csc.kth.se/visgrab

\(^3\)A movie illustrating the hybrid real-world and simulated setup is available at http://cvisl.mmmi.sdu.dk/videos/visgrabCompressed.mp4

\(^4\)http://www.iarm.rra.uka.de/ObjectModels

\(^5\)http://www.robwork.dk
Figure 13: The two grippers used in the experimental evaluation. Left: the Schunk dexterous hand, and right: the Schunk PG70 parallel gripper.

Figure 14: Experimental setup with the parallel jaw gripper.

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. This is the case when the fingers are not in contact with the object after the grasping action has finished.

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

The grasp is considered successful when the returned result is either object slipped or stable grasp. In both cases, the object is held in the hand after lifting.

All three grasp-generation methods introduced in Sect. 4 have been tested on the complete benchmark. The results are given in Sect. 7.1.

6.2 Real-World Setup – Parallel Jaw Gripper

Fig. 14 gives an overview of the real-world setup using the parallel jaw gripper. The setup consists of a Bumblebee2 color stereo camera, an industrial six degrees of freedom Staubli RX60 robot arm, a Schunk PG70 2-Finger Parallel Gripper, and a force-torque sensor that is mounted between the robot’s wrist and the gripper in order to detect collisions.

The three main EGA types using two-fingers, i.e., $c_2$EGA, $sb_2$EGA, and $ss_2$EGA, have been tested. Since this real setup does not allow us to perform anywhere near the number of grasps in the hybrid setup, we decided to not distinguish the different subtypes, but to focus on the main EGA types instead, in order to have enough grasps to calculate reliable averages.

The nine objects used in the experiments are shown in Fig. 15. Approximately 10 different scenes were created for each of the single objects. The scenes were accurately reconstructed in order to test all grasping methods under identical conditions. We thus performed a total of 258 grasping trials with single-object scenes. Additionally, 56 mildly cluttered scenes containing 3 objects were used, see Fig. 16. Again, the scenes have been accurately reconstructed to test the three methods under the same conditions. This resulted in a total of 168 grasping trials with the mildly cluttered scenes.

A grasp is selected at random from the set of grasp hypotheses generated by the given method for a given scene. The outcome of that grasp is labeled as successful when the targeted object is grasped and held in the hand after lifting. In cases...
where another object is lifted as well, due to stacking, the grasp is considered to be successful as well. If the targeted object was dropped, missed, or if a collision occurred, the grasp was labeled as failed.

The results of these experiments are given in Sect. 7.2.

6.3 Real-World Setup – Three-Finger Dexterous Hand

The hardware setup used for the experiments with the three-finger hand consists of a six degrees of freedom industrial robot arm Universal Robot (UR5), a static Bumblebee2 color stereo camera, a force torque sensor mounted at the robot’s wrist, and a Schunk Dexterous Hand (SDH) mounted on the Force Torque sensor, see Fig. 17. The SDH is able to perform both two-finger parallel grasps, as well as three-finger grasps.

In this experiment, six objects have been used, as shown in Fig. 18. The set of objects include well-textured and weakly-textured, as well as open and closed objects. Three of the objects (number 1, 2, and 5) have also been used in the hybrid experimental setup. The other three objects (number 3, 4, and 6) have also been used in the real-world setup using the parallel jaw gripper. This allows us, on the one hand, to compare the real setup with the hybrid setup, and, on the other hand, to compare the two different grippers in the real setups.

Using this setup, the different grasping methods have been tested, that is, the three two-finger and one three-finger grasping methods. As in the previous real-world setup, we do not distinguish the different subtypes, but focus on the main EGA types instead. We performed 120 grasping trials for single-objects. Five different scenes were produced for each object, three where the object was standing up and 2 where the object was laying down, see Fig. 19. Each scene was reproduced four times, once for each grasping method.

For the two-object scenes we tested 72 grasps. The six objects were grouped in three pairs, and each pair was tested withing six scenes, where in three scenes objects were close together and in the remaining scenes objects were far apart. As in the single-object case, each scene was reproduced four times, once for each grasping method.

Identically to the hybrid setup, the robot attempts to grasp the object, and if the hand settles in a stable configuration, it attempts to lift the object. We also adopt the same grasp classification as outlined in Sect. 6.1, which here is made by the experimenter.

The results of these experiments are given in Sect. 7.3.

7 Results

The results of the experiments are given below. We also discuss and compare the results from the different setups and scenarios. The hybrid setup gives the possibility to test a large number of grasps. It also gives the possibility to test all the suggested grasps for one scene. In the case of the real-world experiments, only one of the suggested grasps can be tested per scene since in general the scene is disturbed by the performed grasp attempt. The large number of experiments performed in the hybrid setup provides the in-depth view of the performance of grasping methods and their sub-types, while the evaluation
from the real-world experiments discriminates only between the general grasping methods.

7.1 Hybrid Real-World and Simulated Experimental Setup

The results for the hybrid real-world and simulated experimental setup are split up into three parts. In Sect. 7.1.1, we give the distribution of grasp results. The grasp success as a function of the number of grasp attempts and the combination of the different grasp-generation methods is demonstrated in Sect 7.1.2. Finally, in Set 7.1.3, we show that the different methods are complementary.

7.1.1 Distribution of Grasp Results

Fig. 20 shows the grasp results for the hybrid setup, introduced in Sect. 6.1. The bar plots give the distributions of all grasps averaged over the scenes. The results are split up for the different conditions: single objects standing up, single objects laying down, two objects close together, and two objects far apart. Results are labeled with nograsp when a method does not gen-
ods perform better than the contour-based methods. The en-
three different poses within a pose condition of a scene,
average standard errors on the proportion of successful grasps
rate any grasps for a given scene. In addition, the consistency
erate any grasps for a given scene. In addition, the consistency
the pinch grasps. The three-finger surface-boundary grasp,
perform better than the contour-based methods. The en-
compassing grasps, i.e., EGA1 grasps, are more successful than
All grasping methods perform better when the single objects
are standing upright than when they are laying down. For the
double-object scenes, the amount of successful grasps is simi-

It can be seen that, in general, the surface-based grasp meth-
ods perform better than the contour-based methods. The en-
compassing grasps, i.e., EGA1 grasps, are more successful than

Figure 21: Grasp results for the hybrid real-world and simulated experiments. The plots show the average success rate as a function of the number of grasp attempts for the different grasp methods. The black line shows the performance when the different methods are combined. The fact that the combined results surpass the results of the individual methods shows that they are complementary.
lar for the objects apart or close together, but in the later case, there is a higher proportion of collisions.

On a large proportion of scenes, the contour-based encompassing grasp, $c_2$EGA1, does not suggest any grasps. This is due to the low number of detected contours and the strict requirements of two contours to be parallel and the contact points to be opposing.

For all conditions, the side pinch grasps, i.e., EGA3 grasps, result in more collisions compared to the related encompassing grasps. This is caused by the lack of structural information at the backsides of the object due to self occlusions, which has the consequence that the grasps are not filtered out by the collision-detection mechanism. This is especially the case for the double-object scenes. For the laying objects, however, we do correctly filter out most of the pinch grasps, leading to the large number of scenes where no grasps are suggested. This is because the height of the objects is less when laying down, which reduces the size of the self-occluded part of the object, making collision detection more successful.

The vast majority of collisions are with the target objects, caused by the lack of information about the backsides, due to self-occlusions. Suggested grasps where the hand collides with unobservable parts of the objects are therefore not filtered out by the collision-detection mechanism. In case of the double-object scenes, this occurs more frequently since the suggested grasps can collide with the unobservable parts of the other object as well. In addition, one object can partially occlude the other, causing even more missing information. Collision with the table plane never happened, because the estimation of the table plane was accurate, hence grasps that would cause a collision with the table were correctly filtered out.

### 7.1.2 Grasp Success as a Function of the Number of Grasp Attempts

The average grasp success rates for the different methods lie between 0.2 and 0.6. Although we believe that this is a high performance for a system that does not make use of any object knowledge, it does mean that the robot will often not be able to grasp an object at the first attempt. Assuming that the robot can make a new grasp attempt after the previous grasp fails, we investigate how the success rate increase as a function of the number of grasp attempts. Since we do not order the set of suggested grasps hypotheses, we iteratively select the next grasp at random from the set. In the analysis, we look at the performance per EGA type, as well as the combined performance, where the grasps selected for the different EGA types are taken together. With the combined performance, we investigate if the different grasp types are complementary, i.e., if using more then one grasp type will increase the success rates.

Fig. 21 shows the grasp success as a function of the number of grasp attempts, $N$. We define it as success if any of the grasp attempts is successful. The plots show the average success rate for all scenes over 20 randomized runs. All curves indicate that the success rate increases when multiple grasps are attempted. The steeper the slope, the more successfully additional attempts can be employed. Especially of interest is the black solid line, which shows the grasp success when the different elementary grasping actions are combined. For a single attempted grasp, the success rate for the combined method is the average of the individual EGAs. However, when the eight grasps of each of the EGAs are combined, that is at $N=8$, the curve surpasses the curves of the individual methods in all cases. This shows that the different methods are complementary and that there is a benefit from having different elementary grasping actions based on different types of visual information.

The plots in Fig. 21 also confirm some of the results given in Sect. 7.1.1. The surface-based grasp (orange, red, and green) outperform the contour-based grasps (blue). For the surface-based grasps, the encompassing grasps (solid lines) have a higher success rate than the pinch grasps (dashed and dotted). The contour-based encompassing grasp (solid blue), however, gains little from multiple grasp attempts, which is caused by the low number of grasp suggestions. The three-finger contour-boundary encompassing grasp (solid red) outperforms its two-finger counterpart (solid orange) and is the most successful grasp type overall.

Although for the first grasp attempt, the surface grasps based on the SVD method (green) have a similar success rate to those based on the boundary method (orange), the later is superior when more grasps are attempted. This can partly be explained by the larger number of suggested grasps using the boundary method.

Despite having lower average scores, the contour-based method contributes to the overall system by complementing the surface-based methods when objects are low textured and surface hypotheses based on texlets and surflings cannot be made.

Given the sparse representations of the scene and the heuristics for grasp selection, the grasp-generation methods suggest only a small number of grasps. On average 5-20 grasps are generated per scene for each of the methods. And as can be seen in Fig. 21 good grasp results are generally achieved already after a few attempts, especially when the different grasp methods are combined. The combination of sparseness, complementarity, and high performance is the main contribution of our method.
7.1.3 Complementary nature of the methods

The previous analysis suggests that the different grasp-generation methods complement each other. To get more insight in the contribution of the methods, we perform an additional analysis, the results of which are shown in Fig. 22. The stacked-bar plot shows the rank distribution for the different methods. It shows how often a method is the best performing method, second best, third best, etc., where the methods are ranked based on the success rates (proportion of stable and slipped grasps). The plot confirms that some methods perform better than others, but importantly, it shows that all methods are among the best performing for a proportion of the scenes. All except for ss2EGA1 are even ranked first sometimes. This clearly shows that the methods complement each other and it explains the improved performance when the methods are combined, as shown in the previous section.

7.2 Real-World Setup – Parallel Jaw Gripper

The results of the real-world setup – parallel jaw gripper experiments are shown in Fig. 23. The stacked bars give the distribution of grasp results per object for the different grasp-generation methods. The top figure shows the results for the single-object experiments, and the bottom figure shows the results for the mildly cluttered scenes experiments. The numbers of included grasping trials are given between brackets above the bars.

The plots illustrate the complementary nature of the different grasping methods. For some of the objects the contour-based method performs better, whereas for others, the surface-based methods score best.

Both results from the single-object and mildly cluttered scenes indicate that the contour method is the most successful method in this scenario. This seems to contradict the results from the hybrid setup as given in Sect. 7.1, but can be explained by the shape of the objects and the small aperture of the gripper. Most of the objects in this scenario pose an opening (Fig. 15), and can often only be grasped by the parallel gripper using a top pinch grasps on the sides of the opening. The contours extracted from the top surface of an object often suggest good grasps. Due to the small maximum distance between the two fingers (68 mm), encompassing grasps are often not possible. In contrast, the majority of the objects used in the hybrid setup (Fig. 11) are closed and can not accommodate pinch grasps. Moreover, the SDH gripper used in the hybrid setup allows a much larger aperture, resulting in many successful encompassing grasps on the objects.

Another explanation of the of the worse performance of the
surface-based grasps is that this experimental setup contains fewer textured objects compared to the hybrid experimental setup. It can be observed from this results and the results given in Sect. 7.3 that contour-based grasps are more successful for the weakly textured objects.

For two of the objects in the mildly cluttered scenes, ‘chocolate flakes’ and ‘container angel’, no successful grasps were produced. This is due to the relatively small number of grasp trials for these objects. Since a random grasp is selected for each of the mildly cluttered scenes, some objects were grasped less often than others.

In general, the experiments show an overall performance comparable to the results gained from the simulation. The system is able to successfully grasp unknown objects, both for the single-object and for the mildly cluttered scenes, using the two-finger parallel jaw gripper. In order to test and use all possible grasps for the given size of objects, a bigger gripper is necessary.

As in the hybrid experimental setup, also here, the vast majority of collisions are with the target object caused by missing information due to (self-)occlusions. Additionally, a very small number of incidental collisions with the table occurred.

7.3 Real-World Setup – Three-Finger Dexterous Hand

Fig. 24 shows the outcomes of different grasping methods for individual objects. It can be seen that for the weakly textured kitchen container, the contour method is the only method that produces stable grasps. Also for the weakly textured red basket, the contour grasps perform best, with over 80% success when both stable and less stable grasps are taken into account. Both in the case of the kitchen container and the red basket, a large amount of collisions occurred, due to the fact that the object surfaces are not detected.

For textured objects, the success of different methods varies across objects, which once more confirms the supplementary strength of different methods, as seen in the previous results from Sect. 7.1 and Sect. 7.2. The difference compared to the results from the hybrid setup is that the three-finger sb3EGA1 does not perform as well. The sb3EGA1 is an encompassing grasp of the whole surface and this fits well with the objects used in the simulated setup, objects that were mostly closed, with well-textured top surface. Two such objects from the real experiments are corny and marmalade, and the increase in the performance of the sb3EGA1 method for the two objects can be seen in Fig. 24.

In Fig. 25 (top), the outcomes of different grasping methods are averaged over all objects. The success rate of the first grasping attempt varies between 30-60%, in agreement with results from the hybrid setup. Also similar to the hybrid setup, the results for the objects in the up-right position are better than for the objects laying down. The results for the two object scenes are better when the objects are far apart, see Fig. 25 (bottom).

Also in this experimental setup, virtually all collisions are with the target object caused by lack of information due to (self-)occlusions. Note, however, that the amount of collisions is somewhat smaller here than in the hybrid experimental setup. This can be explained from the fact that in the simulation, the hand can be freely place at any position and with
any orientation. In the real setups, on the other hand, the hand pose is restricted by the robotic arm. Since the stereo cameras are placed at the same side of the objects as the arm, the invisible sides, where there is a higher risk of collision, are not reachable, and thus not considered for grasping.

Three of the objects used in these real experiments have also been used in the hybrid setup (coke bottle, corny, and marmalade). Figure 26 shows the results for these objects in the hybrid experimental setup. Remember that different from the real experiments, the EGAs are split up in subtypes. The results for the coke bottle and the marmalade pot are very similar in the real and hybrid setup. For the corny box, the results are somewhat different, especially for the contour-based EGA, which performs much better in the real setup than in the hybrid setup. In general, the contour-based method shows slightly better performance in the real scenario. This can be explained by the homogeneously colored white table, which simplifies the detection of edges and contours.

8 Discussion

In this paper, we presented a bottom-up vision system for general scene interpretation in terms of hierarchies of visual descriptors, and we used this system for grasping unknown objects. We continued our earlier work (Popović et al., 2010), by extending the hierarchical representation of the Early Cognitive Vision (ECV) system to the texture and surface domain, and by using the representation to generate not only contour-based, but also surface-based grasps.

The ECV system organizes multi-modal three-dimensional visual information in a hierarchy of increasing level of abstraction. In the contour domain, local edge features on the lowest level are grouped into contours, and contours belonging to the same surface are further grouped together. In the texture domain, local textured patches on the lowest level are grouped into larger surfings, which are then grouped to form surfaces. This approach has four advantages: 1) the Gestalt principles used in the grouping process result in a noise-robust scene representation, 2) it allows us to address the grasping problem at a level in the hierarchy which sufficiently narrows the search space for grasping possibilities, 3) the vision system extracts rich visual information about surfaces of objects in the scene, which allows to extract grasping hypotheses, and 4) by using lower levels of hierarchy we can define good contact points. We proposed different elementary grasping actions, utilizing contour and surface information present in the hierarchical representation made of the scene.

We tested the system in different experimental setups. First, we used a hybrid real-world and simulated setup. Based on real stereo images, our method built a visual representation of the scene and generated grasps. These grasps were then executed in a dynamic simulator. This setup allowed us to test a large number of grasps and get quantitative results, while still dealing with the noise and uncertainty in the real-world visual data. It allowed us, moreover, to compare the different methods under the exact same circumstances. Second, to show the applicability of the proposed methods on real robotic systems, we tested the proposed methods on two different real-world experimental setups, using a parallel and a three-finger dexterous hand.

The results show a good average performance of the proposed grasping methods, which is even further improved when the different methods are combined. This shows that the different methods, which are based on different types of visual information and apply different grasps to contact points, complement each other so that a better performance can be achieved. In particular, the contour-based methods perform better in the situations where objects are not textured, while different surface-based methods complement each other in the situations where objects are textured. By combining both types of information, the overall system can deal with both types of objects.

We proposed two surface-based methods. The SVD method uses a rectangular approximation of the surface, and therefore suggests a limited number of grasps. The boundary method suggests a larger number of grasps, based on a more general representation of the surface, which can deal with a larger variety of shapes.

If we compare the overall results of the single-object scenes with the complex scenes, we see that the success rates are in the same range. This shows that our hierarchical vision system provides a powerful representation of the scene that can be used to generate good grasps, even with increasing visual complexity. Similarly, in the hybrid setup, the results for the scenes with textured and non-textured background are similar, which also indicates that the proposed methods are robust to different levels of visual complexity.
The experiments from the hybrid setup provided an in-depth analysis of the grasping methods and their sub-types. The experiments from the two real-world setups confirmed that the same level of success can be expected when grasping in real-world scenarios. Performing the experiments on the different setups and with different objects has helped to better understand the proposed grasping methods.

Since we are dealing with grasping unknown objects in unknown scenes, a 100% success rate is not to be expected. However, the results show that a good average performance can be achieved and that the grasp success strongly increases when more attempts are made and different grasp types are combined. Attempting multiple grasps might not always be desirable as it can disturb the scene or damage the object. However, the proposed method can easily be combined with our previous work on grasp-stability assessment based on haptic feedback (Bekiroglu et al., 2011), which can be used to detect an unstable grasp without actually lifting the object. This allows to regrasp the object without disturbing the scene.

The different EGAs make different assumptions about the scene. Our future work aims to develop the ECV system on two issues, in order to improve reasoning about these assumptions. First, the current representation cannot be used to predict collisions of the gripper with the unobservable parts of objects due to (self-)occlusions. This is an inherently difficult problem in grasping unknown objects. The current system utilizes a semi-global representation of the objects in terms of its surfaces. To better deal with occlusions, we propose to extend the ECV system to the object level and not only rely on bottom-up processes, but to also let the system make estimations about the global shape of the object. Back-projecting these estimations to lower levels will enable hallucination of the back sides of objects. Second, the amount of cross-modal interaction between contour and surface information is currently limited. Surface information is used for collision detection to filter contour-based grasps, but otherwise the two modalities are independent. By combining information from both sources, our future system will have an even more complete and powerful representation of the scene.

The main focus of the work presented in this paper is on the individual grasp-generation methods. The selection of grasps and the combination of methods are therefore quite naive. To improve performance, we will target these issues in future work. We will pursue a developmental approach by mechanisms that learn from the system’s grasp experience, such as proposed in, e.g., (Kraft et al., 2008; Natale et al., 2005). This will enable the system to improve grasp performance over time by predicting the grasps quality for more efficient grasp selection and by selecting the appropriate grasp type.

We want to stress that once a system is equipped with an innate grasping behavior enabling it to gather grasp experience with sufficient number of successful grasps, associations between sensory information and successful actions can be learned. As we show in this paper, it is possible to create such innate mechanisms based on ECV features. Using the grasp experience, it is then possible to formalize the problem of learning new and even more efficient feature-grasp associations compared to the ones defined in the innate mechanisms. The hierarchical representation of the ECV system allows us to define this learning problem on a suitable level of granularity to make the learning task feasible.

The VisGraB benchmark that we used in this paper is open for scientific use. The stereo images, the simulated environment, and the dynamic simulator are available on http://www.csc.kth.se/visgrab.

Acknowledgments

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

References


C. S. Curtis and J. Xiao. Efficient and effective grasping of novel objects through learning and adapting a knowledge base. In


