# Vision-based Robotic Grasping from Object Localization, Pose Estimation, Grasp Detection to Motion Planning: A Review

**Guoguang Du** , **Kai Wang** and **Shiguo Lian**
Cloudminds Technologies
{george.du, kai.wang, scott.lian}@cloudminds.com

## Abstract

This paper presents a comprehensive survey on vision-based robotic grasping. We concluded four key tasks during robotic grasping, which are object localization, pose estimation, grasp detection and motion planning. In detail, object localization includes object detection and segmentation methods, pose estimation includes RGB-based and RGB-D-based methods, grasp detection includes traditional methods and deep learning-based methods, motion planning includes analytical methods, imitating learning methods, and reinforcement learning methods. Besides, lots of methods accomplish some of the tasks jointly, such as object-detection-combined 6D pose estimation, grasp detection without pose estimation, end-to-end grasp detection, and end-to-end motion planning. These methods are reviewed elaborately in this survey. What's more, related datasets are summarized and comparisons between state-of-the-art methods are given for each task. Challenges about robotic grasping are presented, and future directions in addressing these challenges are also pointed out.

## 1 Introduction

An intelligent robot is expected to not only be able to perceive the environment, but also interact with the environment. Among all these abilities, object grasping is fundamental and significant in that it will bring enormous productivity to the society [Sanchez *et al.*, 2018]. For example, an industrial robot can accomplish the pick-and-bin task which is laborious for human labors, and a domestic robot is able to provide assistance to disabled people in their daily grasping tasks.

In order to accomplish the robotic grasping task, a robot needs to perceive the objects first. With the development of sensor devices, robots nowadays are equipped with RGB cameras as well as depth cameras to capture the rich information of the environment. However, raw RGB-D images are simple grids of numbers to the robot, where high-level semantic information should be extracted to enable vision-based perception. The high-level information of the target object to grasp usually contains the location, the orientations, and the grasp positions. The grasp planning are then computed to execute the physical grasp. Endowing robots with the ability to perceive has been a long-standing goal in computer vision and robotics discipline.

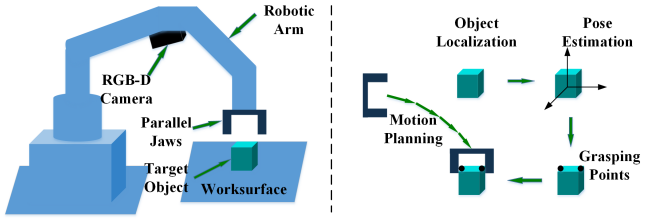


Figure 1: The grasp detection system. (Left)The robotic arm equipped with the RGB-D camera and two parallel jaws, is to grasp the target object placed on a planar worksurface. (Right)General procedures of robotic grasping involves object localization, pose estimation, grasping points detection and motion planning.

As much as being highly significant, robotic grasping has long been researched. The robotic grasping system is considered as being composed of the following sub-systems [Kumra and Kanan, 2017]: the grasp detection system, the grasp planning system and the control system. Among them, the grasp detection system is the key entry point, as illustrated in Fig. 1. It is divided into three tasks: target object localization, pose estimation and grasp point detection. Together with grasp planning, the four tasks will be introduced elaborately. As the control system is more relevant to the automation discipline, it will not be included in this survey.

Among all the tasks mentioned above, there have been some works [Sahbani *et al.*, 2012; Bohg *et al.*, 2014; Caldera *et al.*, 2018] concentrating on one or a few tasks, while there is still a lack of comprehensive introduction on robotic grasping. To the best of our knowledge, this is the first review that broadly summarizes the progress and promises new directions in vision-based robotic grasping of 3D objects. We believe that this contribution will serve as an insightful reference to the robotic community.

It should be mentioned that for the localization and the pose estimation tasks, we focused on the small objects that could be grasped by a robotic arm, while those of other objects like human body, vehicles, etc., will not be discussed here. For the grasp detection task, we only considered the

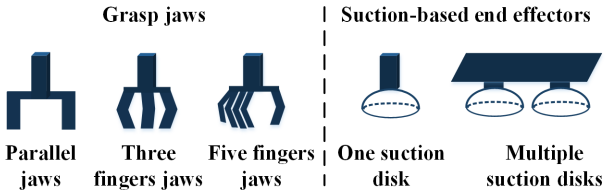


Figure 2: Different kinds of end effectors. (Left)Grasp jaws. (Right)Suction-based end effectors. In this paper, we only consider parallel jaws.

robotic arm with two parallel fingers, as shown in Fig. 2.

The remainder of the paper is arranged as follows. A brief history of robotic grasping will be introduced in Chapter 2. In Chapter 3, representative works will be categorized into eight methods and discussed in detail. Next, the datasets relating with the four tasks, the evaluation metrics and comparisons about state-of-the-art methods are presented in Chapter 4. Finally, the challenges and future directions are summarized in chapter 5.

## 2 A Brief History of Robotic Grasping

Robotic grasping has long been researched in literature. In early years, the robotic arm was not endowed with the perception ability, and robotic grasping was conducted by manually controlling the mechanical arms and hands. Later on, with the assistance of tactile sensors such as data gloves, a robot can grasp by coping the behaviour of human hands. Both of these methods still rely on human labor and are with few intelligence.

With the development of optical sensors, robots have been able to grasp automatically through vision-based perception system. Traditionally, a vision-based robotic grasping system is composed of a series of components, including target object localization, object pose estimation, grasp detection and grasp planning [Sahbani *et al.*, 2012]. Object localization finds the location of the target object in the input data. Object pose estimation estimates the rotation as well as the translation of the target object with respect to a reference. Grasp detection computes the grasp configuration with regard to the target object. This configuration can be represented as a seven dimensional variable, including a grasping point, grasping orientation, and gripper opening width [Jiang *et al.*, 2011]. Grasp planning refers to the path planning process which is required to safely grab the object and maintain the closed gripper contacts to hold and lift the object from its resting surface [Bicchi and Kumar, 2000].

Early methods assume that the object to grasp is placed in a clean environment with simple background and thus simplifies the object localization task, while in relatively complex environments their capabilities are quite limited. Some object detection methods utilized machine learning methods to train classifiers based on hand-crafted 2D descriptors. However, these classifiers show limited performance since the limitations of hand-crafted descriptors. In recent years, deep learning has begun to dominant the image-related tasks such as object detection and segmentation. Besides, the training data ranges from RGB images to depth images, and deep learning networks with 2D or 3D inputs are proposed, which highly improves the performance of object localization and extremely prompts the development of robotic grasping.

With the location of the target object, the grasp detection can be conducted. In early years, analytical methods were utilized which directly analyze the geometric structure of the input data, and find the points suitable to grasp according to force closure or shape closure. Sahbani et al. [Sahbani *et al.*, 2012] presented an overview of 3D object grasping algorithms, where analytical approaches are introduced in detail. However, analytical methods have many problems such as time consuming, difficult to compute the force. Later, with the emergence of large numbers of 3D models, data-driven methods could be analyzed to transfer grasps in the 3D model database to the target object. Bohg et al. [Bohg *et al.*, 2014] reviewed data-driven grasp methods and divided the approaches into three groups based on whether the grasp configuration is computed for known, familiar or unknown objects. Generally, the 6D pose of the target object is essential to accomplish this task. Both RGB image-based and depth image-based methods could achieve accurate pose estimations. However, these methods such as partial registration methods [Besl and McKay, 1992a] are susceptible to sensor noise or incomplete data. The poses could also be estimated directly or indirectly from the input data through deep learning methods in order to get resistance to the sensor noise or incomplete data. There also exist deep learning-based methods where the 6D poses are not needed to conduct grasp detection. The grasp configuration could be regressed directly or indirectly through deep convolutional networks. Caldera et al. [Caldera *et al.*, 2018] reviewed the deep learning-based methods in robotic grasping detection. They discussed how each element of the deep learning approach improves the overall performance of robotic grasping detection. Besides, the supervised learning methods, reinforcement learning have also been utilized to directly accomplish specific tasks like toy assembly and pour water, which are closely related to grasping.

## 3 Methods Overview

The four tasks in vision-based robotic grasping can be accomplished either independently or jointly, and eight methods can thus be categorized, as shown in Fig. 3. Each kind of methods can accomplish one or more tasks. Method 1, method 2, method 4 and method 7 can accomplish object localization, pose estimation, grasp detection, and motion planning, respectively. In method 3, object localization and pose estimation are accomplished together. Methods 5 can perform grasp detection without estimating the 6D pose of the object. For method 6, grasp detection is completed without object localization and pose estimation. Method 8 accomplishes the whole grasp task directly from the input data. Next, these eight methods will be discussed in detail.

### 3.1 Object localization

Most grasping approaches require the computation of the target object's location in the input image data first. This involves object detection and segmentation techniques. Object

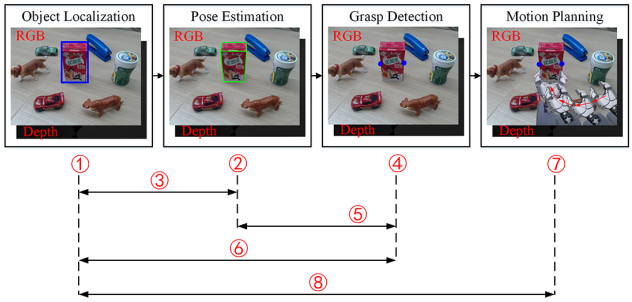


Figure 3: The proposed classification of current methods. Eight kinds of methods are involved by separately or jointly accomplishing the robotic grasping task.

detection provides the rectangular bounding box of the target object, and object segmentation provides the precise boundary of the target object. The latter provides more accurate descriptions of the object area, while its computation is more time consuming. The representative works of the two methods are shown in Table 1.

Table 1: Summary of object localization methods.

| Method | Known Info | Key idea | Representative methods |
|---|---|---|---|
| Object detection | 2D object detection | Use manually designed descriptors or deep learning-based methods | SIFT [Lowe, 1999], SURF [Bay *et al.*, 2006], Bag of Words [Galvez-López and Tardos, 2012], DCNN [Krizhevsky *et al.*, 2012], RCNN [Girshick *et al.*, 2014], Mask RCNN [He *et al.*, 2017], YOLO [Redmon *et al.*, 2016], SSD [Liu *et al.*, 2016] |
| | 3D object detection | Use 3D shape descriptors | Spin image [Johnson, 1997], FPFH [Rusu *et al.*, 2009], SHOT [Salti *et al.*, 2014] |
| Object segmentation | 2D object segmentation | Using clustering methods or deep learning-based methods | Long et al. [Long *et al.*, 2015a], SegNet [Badrinarayanan *et al.*, 2017], DeepLab [Chen *et al.*, 2018] |
| | 3D object segmentation | Fitting 3D primitives or deep learning methods | PointNet [Qi *et al.*, 2017], PointCNN [Li *et al.*, 2018b] |

## Object detection

Typical functional flow-chart of 2D object detection is illustrated in Fig. 4. Traditional 2D object detection methods rely on template matching, which utilized manually designed descriptors, such as SIFT [Lowe, 1999], SURF [Bay *et al.*, 2006], Bag of Words [Galvez-López and Tardos, 2012] and so on. Researchers trained classifiers, such as neural networks, support vector machine or Adaboost, according to the descriptors to conduct object detection. Although the descriptors have been widely used in various vision-related tasks, deep learning-based methods have become popular in recent years since the proposal of deep convolutional neural network(DCNN) [Krizhevsky *et al.*, 2012]. These methods can be further divided into two-stage methods and one-stage methods. Two-stage methods include the pre-processing for region proposal, making the overall pipeline two stages which are region proposals generation and ranking of the best. One-stage methods utilize a unified pipeline to output the detection results directly and skip the separate proposal detection. Representative works of two-stage methods include RCNN [Girshick *et al.*, 2014], Mask RCNN [He *et al.*, 2017] and one-stage methods include YOLO [Redmon *et al.*, 2016], SSD [Liu *et al.*, 2016], etc. The YOLO detection system is illustrated in Fig. 5. Detailed review of these works please refer to a recent survey [Liu *et al.*, 2018].

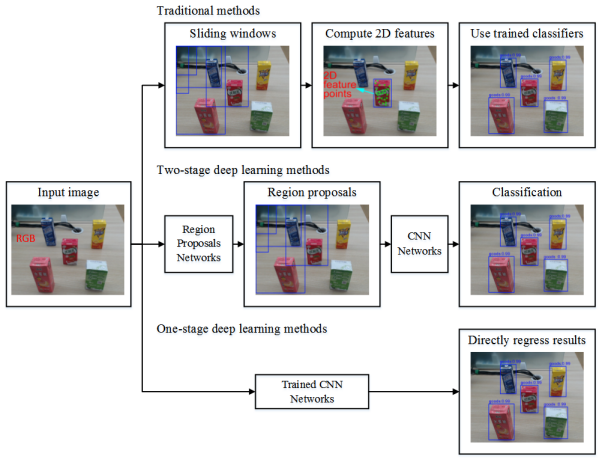


Figure 4: Typical functional flow-chart of 2D object detection.

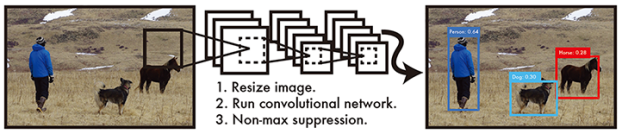


Figure 5: The YOLO detection system. Processing images with YOLO is simple and straightforward. The system resizes the input image to $448 \times 448$, runs a single convolutional network on the image, and thresholds the resulting detections by the model's confidence. (Courtesy of [Redmon *et al.*, 2016])

Typical functional flow-chart of 3D object detection is illustrated in Fig. 6. Except the RGB image data, the depth image can also be used to conduct the detection task by mapping to 3D point cloud and using 3D local shape descriptors, such as FPFH [Rusu *et al.*, 2009], SHOT [Salti *et al.*, 2014], etc. Various deep learning-based 3D object detection methods [Simon *et al.*, 2018; Ali *et al.*, 2018; Yang *et al.*, 2018; Song and Xiao, 2014; Ren and Sudderth, 2018] on 3D data are also proposed in recent years. The framework of the Complex-YOLO method is illustrated in Fig. 7.

## Object segmentation

Typical functional flow-chart of 2D object segmentation is illustrated in Fig. 8. Traditional segmentation methods conduct the computation based on clustering methods or graph cut-based methods, while their performance is still limited. Long et al. [Long *et al.*, 2015a] successfully adopted the deep neural network into image segmentation, as shown in Fig. 9, and a series of following works, such as SegNet [Badrinarayanan *et al.*, 2017] and DeepLab [Chen *et al.*, 2018] were further

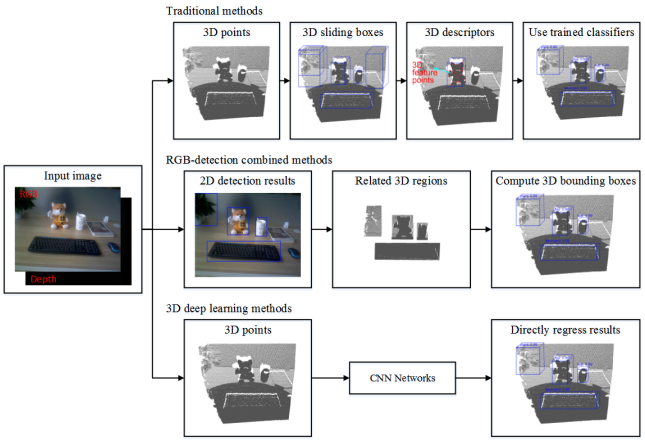


Figure 6: Typical functional flow-chart of 3D object detection.

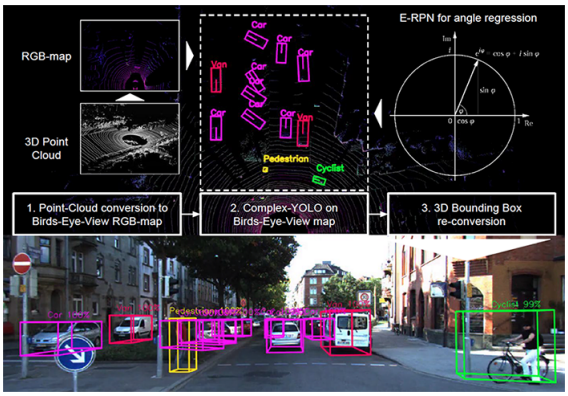


Figure 7: Complex-YOLO directly operates on Lidar only based birds-eye-view RGB-maps to estimate and localize accurate 3D multiclass bounding boxes. The upper part of the figure shows a bird view based on a Velodyne HDL64 point cloud such as the predicted objects. The lower one outlines the re-projection of the 3D boxes into image space. Complex-YOLO needs no camera image as input, it is Lidar based only. (Courtesy of [Simon *et al.*, 2018])

proposed. Detailed reviews refer to the survey [Garcia-Garcia *et al.*, 2017].

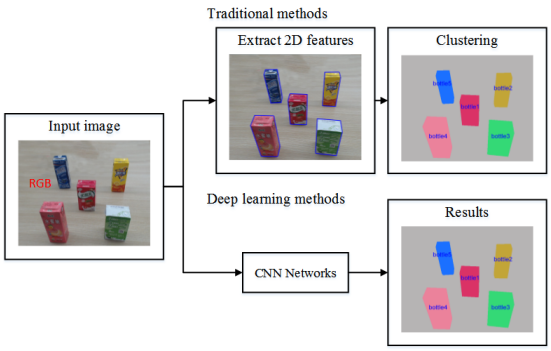


Figure 8: Typical functional flow-chart of 2D object segmentation.

Typical functional flow-chart of 3D object segmentation is

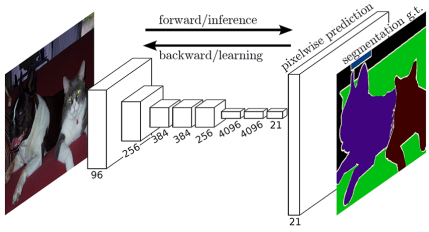


Figure 9: Fully convolutional networks can efficiently learn to make dense predictions for per-pixel tasks like semantic segmentation. (Courtesy of [Long *et al.*, 2015a])

illustrated in Fig. 10. Besides, there also exist 3D segmentation methods. These methods used to utilize clustering methods or fitting primitives [Nguyen and Le, 2013]. Recently, focus has been shifted to using 3D neural networks, such as PointNet [Qi *et al.*, 2017], PointCNN [Li *et al.*, 2018b], etc., to provide more accurate 3D segmentation results. Applications of PointNet are illustrated in Fig. 11.

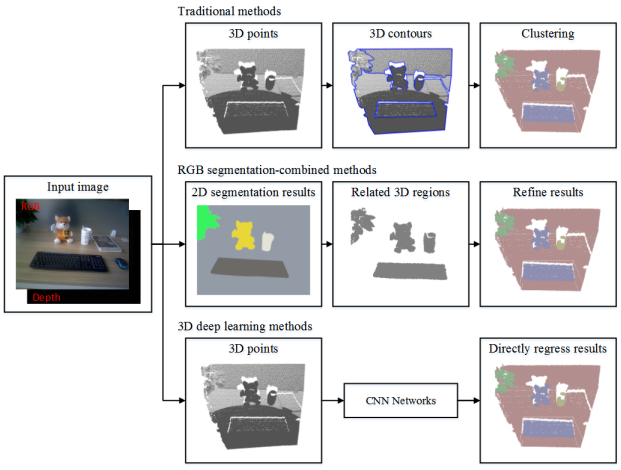


Figure 10: Typical functional flow-chart of 3D object segmentation.

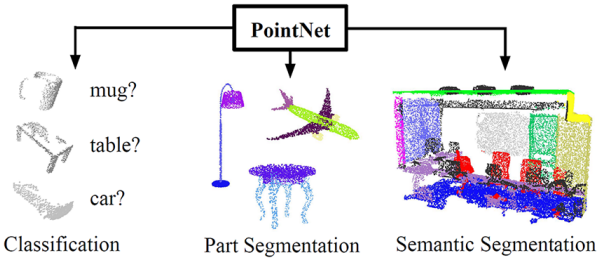


Figure 11: Applications of PointNet which consumes raw point cloud (set of points) without voxelization or rendering. It is a unified architecture that learns both global and local point features, providing a simple, efficient and effective approach for a number of 3D recognition tasks. (Courtesy of [Qi *et al.*, 2017])

## Challenges

Comparing to traditional hand-crafted descriptors-based methods, the deep learning-based approaches achieved bet-

ter results. However, the need for large amounts of training data and the generalization ability of the trained models still remain challenging.

## 3.2 6D Pose Estimation

6D pose estimation plays a pivotal role in many areas such as augmented reality, robotic manipulation, and autonomous driving, et al. It helps a robot to get aware of the position and orientation of the object to grasp. The pose estimation methods can be roughly divided into four kinds, which are based on correspondence, template, voting, and regression respectively, as shown in Table 2. The first three kinds of methods which rely on the detected or segmented objects will be discussed in this section, and the regression-based methods which accomplish object detection and 6D pose estimation together will be discussed in the next section.

Table 2: Summary of 6D pose estimation methods.

| Method | Known Info | Key idea | Representative methods |
|---|---|---|---|
| Correspondence-based method | 3D point cloud, Rendered images from 3D model with texture | Find Correspondences between 2D points and 3D points, and use PNP methods | SIFT [Lowe, 1999], SURF [Bay *et al.*, 2006], ORB [Mur-Artal *et al.*, 2015] |
| | 3D point cloud | Find 3D Correspondence through random hypothesis or 3D descriptors, and results are refined using ICP | Spin image [Johnson, 1997], FPFH [Rusu *et al.*, 2009], SHOT [Salti *et al.*, 2014] |
| Template-based method | 3D point cloud, Rendered images from 3D model with no texture | Extract the gradient information for matching, and results are refined using ICP | LineMod [Hinterstoisser *et al.*, 2012], Hodaň et al. [Hodaň *et al.*, 2015] |
| Voting-based method | 3D point cloud or rendered RGB-D images with pose | Every local predicts a result, and results are refined using RANSAC | Brachmann et al. [Brachmann *et al.*, 2014], PPF [Drost and Ilic, 2012], DenseFusion [Wang *et al.*, 2019a] |
| Regression-based method | 3D point cloud or rendered RGB-D images with pose | Represent pose suitable for CNN | BB8 [Rad and Lepetit, 2017], SSD6D [Kehl *et al.*, 2017], PoseCNN [Xiang *et al.*, 2017], Deep6DPose [Do *et al.*, 2018] |

**Correspondence-based methods**

Typical functional flow-chart of correspondence-based object pose estimation methods without object detection is illustrated in Fig. 12. This kind of method mainly targets on the pose estimation of objects with rich textures for the matching of 2D feature points. Multiple images are firstly rendered by projecting the existing 3D models from various angles. By finding the matchings between 2D feature points on the observed image and the rendered images [Vacchetti *et al.*, 2004], as shown in Fig. 13, the 2D pixel to 3D point correspondences are established. The common 2D descriptors such as SIFT [Lowe, 1999], SURF [Bay *et al.*, 2006], ORB [Mur-Artal *et al.*, 2015], etc., are usually utilized for the 2D feature extraction. The 6D pose of the object can be calculated with Perspective-n-Point(PnP) algorithms [Lepetit *et al.*, 2009]. Lepetit et al. [Lepetit *et al.*, 2005] present a survey about monocular model-based 3D tracking methods of rigid objects.

When the depth image is available, the problem turns into a partial registration problem. The popular 3D descriptors, such as FPFH [Rusu *et al.*, 2009], Spin image [Johnson, 1997], and SHOT [Salti *et al.*, 2014], can be utilized to find correspondences between the partial 3D point cloud and the full object to obtain a rough pose. It is then refined through the iterative closest points(ICP) algorithm [Besl and McKay, 1992b]. The influence region diagram for the FPFH descriptor is illustrated in Fig. 14. This kind of methods rely on the geometry of the target object and are widely used in pose estimation of robotic grasping. Wong et al. [Wong *et al.*, 2017] proposed a method which integrated RGB-based object segmentation and depth image-based partial registration to obtain the pose of the target object. They presented a novel metric for scoring model registration quality, and conducted multi-hypothesis registration, which achieved accurate pose estimation with $1cm$ position error and $< 5^\circ$ angle error.

**Template-based methods**

Typical functional flow-chart of template-based object pose estimation methods without object detection is illustrated in Fig. 15. This kind of methods aims at computing the pose of objects with no textures, for which the correspondence-based methods can hardly be used. In these methods, the gradient information is usually utilized. Multiple images which are generated by projecting the existing 3D models from various angles will be regarded as the templates. Hinterstoisser et al. [Hinterstoisser *et al.*, 2012] proposed a novel image representation by spreading image gradient orientations for template matching and represented a 3D object with a limited set of templates, as shown in Fig. 16. The accuracy of the estimated pose was improved by taking into account the 3D surface normal orientations which are computed from the dense point cloud obtained from a dense depth sensor. Hodaň et al. [Hodaň *et al.*, 2015] proposed a method for the detection and accurate 3D localization of multiple texture-less and rigid objects depicted in RGB-D images. The candidate object instances are verified by matching feature points in different modalities and the approximate object pose associated with each detected template is used as the initial value for further optimization.

**Voting-based methods**

Typical functional flow-chart of voting-based object pose estimation methods without object detection is illustrated in Fig. 17. This kind of methods are mainly used for computing the poses of objects with occlusions. For these objects, the local evidence in the image restricts the possible outcome of the desired output, and every image patch is thus usually used to cast a vote about the output. Brachmann et al. [Brachmann *et al.*, 2014] proposed a learned, intermediate representation in form of a dense 3D object coordinate labelling paired with a dense class labelling. Each object coordinate prediction defines a 3D-3D correspondence between the image and the 3D object model, and the pose hypotheses are generated and refined to obtain the final hypothesis. Tejani et al. [Tejani *et al.*, 2014] trained a Hough forest for 6D pose estimation from an RGB-D image. Each tree in the forest maps an image patch to a leaf which stores a set of 6D pose votes. Drost et al. [Drost *et al.*, 2010] proposed the Point Pair Features(PPF) to recover

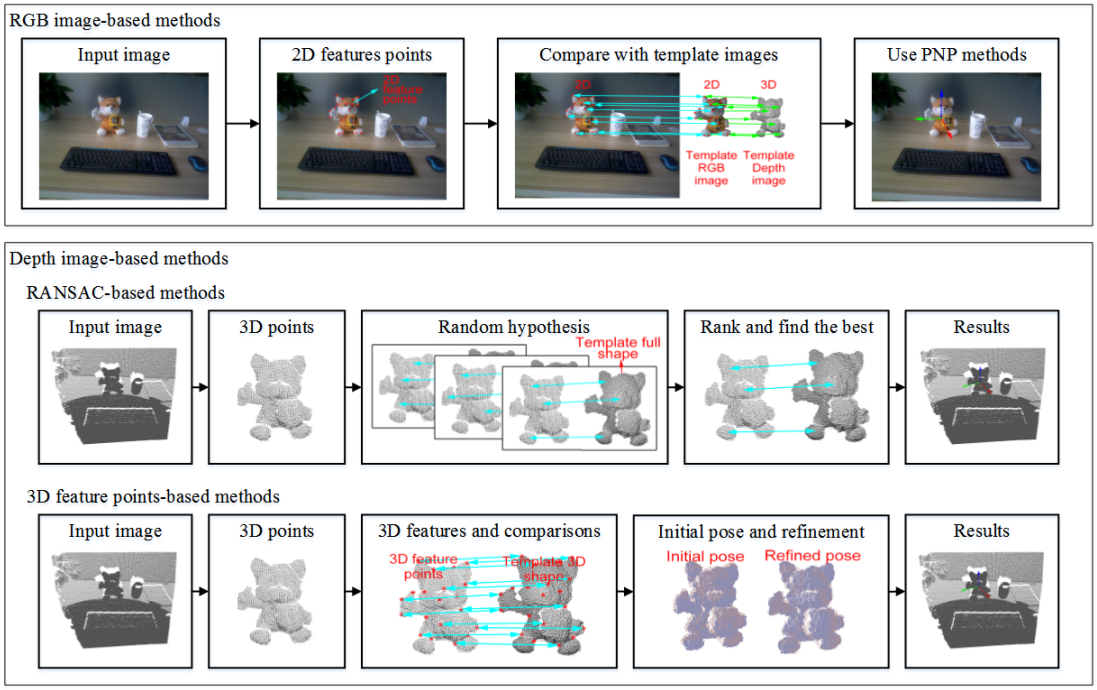


Figure 12: Typical functional flow-chart of correspondence-based object pose estimation methods without object detection.

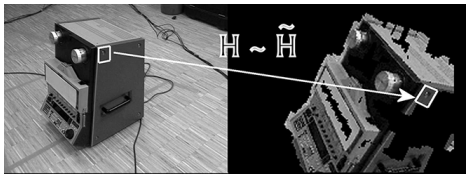


Figure 13: Pixels around interest points are transferred from the keyframe (left) to the re-rendered image (right) using a homography, and 2D-to-3D correspondences are found. (Courtesy of [Vacchetti *et al.*, 2004])

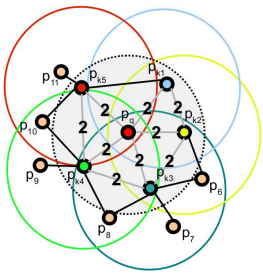

Figure 14: The influence region diagram for a Fast Point Feature Histogram. Each query point (red) is connected only to its direct k-neighbors (enclosed by the gray circle). Each direct neighbor is connected to its own neighbors and the resulted histograms are weighted together with the histogram of the query point to form the FPFH. The connections marked with 2 will contribute to the FPFH twice. (Courtesy of [Rusu *et al.*, 2009])

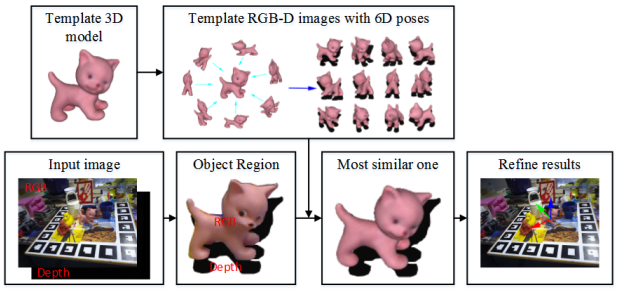


Figure 15: Typical functional flow-chart of template-based object pose estimation methods without object detection. Data from the lineMod dataset [Hinterstoisser *et al.*, 2012].

Figure 16: Template generation and the feature points. Left: Red vertices represent the virtual camera centers used to generate templates. Middle: The color gradient features are displayed in red and surface normal features in green. Right: 15 different texture-less 3D objects used. (Courtesy of [Hinterstoisser *et al.*, 2012])

the 6D pose of objects from a depth image. A point pair feature contains information about the distance and the normals of two arbitrary 3D points, as shown in Fig. 18. PPF has been one of the most successful 6D pose estimation method as an efficient and integrated alternative to the traditional local and global pipelines. Hodan et al. [Hodan *et al.*, 2018a] proposed

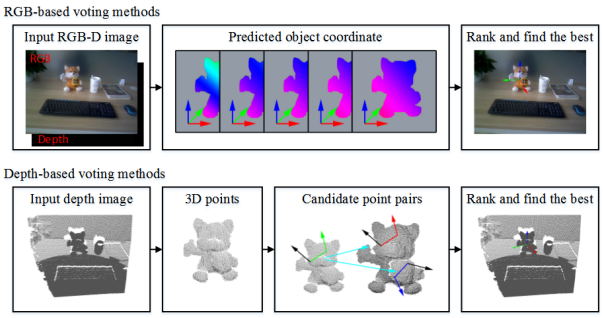


Figure 17: Typical functional flow-chart of voting-based object pose estimation methods without object detection.

a benchmark for 6D pose estimation of a rigid object from a single RGB-D input image, and a variation of PPF [Vidal *et al.*, 2018] won the SIXD challenge.

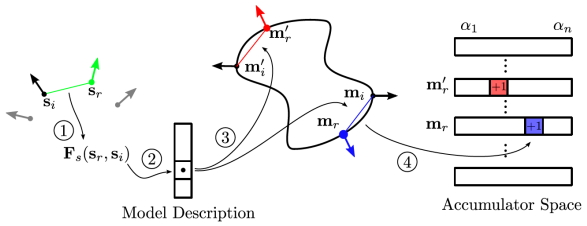


Figure 18: Visualisation of different steps in the voting scheme: (1) Reference point $s_r$ is paired with every other point $s_i$, and their point pair feature $F$ is calculated. (2) Feature $F$ is matched to the global model description, which returns a set of point pairs on the model that have similar distance and orientation. (3) For each point pair on the model matched to the point pair in the scene, the local coordinate $\alpha$ is calculated by solving $s_i = T_{s\to g}^{-1} R_x(\alpha) T_{s\to g} m_i$. (4) After $\alpha$ is calculated, a vote is cast for the local coordinate $(m_i, \alpha)$. (Courtesy of [Drost *et al.*, 2010])

Wang et al. [Wang *et al.*, 2019a] proposed a generic framework named DenseFusion for estimating the 6D pose of a set of known objects from RGB-D images. DenseFusion is a heterogeneous architecture that processes the two data sources (RGB and depth) independently and uses a novel network to extract pixel-wise dense feature embeddings, from which the pose is estimated. The predictions are voted to generate the final 6D pose prediction of the object.

**Challenges**

The main challenge of these pose estimation methods lies in that accurate 3D models are required to obtain accurate results, while in most cases it is difficult to obtain accurate 3D digital models of the target objects. This leads to the proposal of deep learning-based methods which still suffers from the lack of accuracy in generalizing to novel objects.

### 3.3 Object Detection-Combined 6D Pose Estimation

This kind of method is also referred as regression-based method, which accomplishes object detection and 6D pose estimation together, as shown in Fig. 3. Typical functional flow-chart of object detection-combined pose estimation methods is illustrated in Fig. 19. Different from other methods which adopt multi-staged strategies to estimate object pose from input images, this kind of method learns the immediate mapping from an input image to a parametric representation of the pose, and the 6D object pose can thus be estimated combined with object detection [Patil and Rabha, 2018]. The regression-based methods could be divided into two kinds: one directly regress the 6D object pose [Xiang *et al.*, 2017; Do *et al.*, 2018], and the other regress positions of key points which provides 2D and 3D correspondences [Rad and Lepetit, 2017; Tekin *et al.*, 2018].

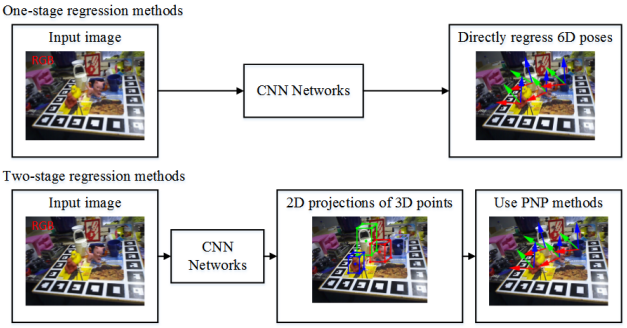


Figure 19: Typical functional flow-chart of object detection-combined pose estimation methods. Data from the lineMod dataset [Hinterstoisser *et al.*, 2012].

The kind of methods which directly regresses the 6D object pose is firstly proposed. Yu et al. [Xiang *et al.*, 2017] proposed PoseCNN, a new convolutional neural network for direct 6D object pose estimation, as shown in Fig. 20. The 3D translation of an object is estimated by localizing the center in the image and predicting the distance from the camera, and the 3D rotation is computed by regressing to a quaternion representation. Besides, they also introduced the ShapeMatch-Loss function that enables PoseCNN to handle symmetric objects. Do et al. [Do *et al.*, 2018] proposed an end-to-end deep learning framework named Deep-6DPose, which jointly detects, segments, and recovers 6D poses of object instances form a single RGB image. They extended the instance segmentation network Mask RCNN [He *et al.*, 2017] with a novel pose estimation branch to directly regress 6D object poses without any post-refinements. Liu et al. [Liu *et al.*, 2019] proposed a two-stage CNN architecture which directly outputs the 6D pose without requiring multiple stages or additional post-processing like PnP. They transformed the pose estimation problem into a classification and regression task.

The second kind of methods conducts an indirectly way to regress the 6D pose, which first computes 2D-3D correspondences and then solves the PnP problem [Lepetit *et al.*, 2009] to obtain the 6D pose. Rad and Lepetit [Rad and Lepetit, 2017] predicted 2D projections of the corners of their 3D bounding boxes and obtained the 2D-3D correspondences. Kehl et al. [Kehl *et al.*, 2017] presented a similar method by making use of the SSD network. Different with them, Tekin et al. [Tekin *et al.*, 2018] proposed a single-shot deep CNN architecture that directly detected the 2D projections of the 3D bounding box vertices without any a posteriori refine-

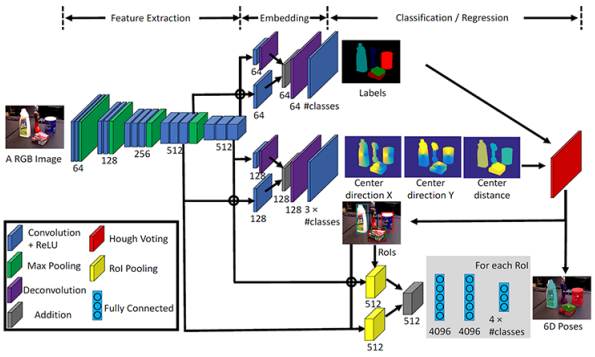


Figure 20: Architecture of PoseCNN for 6D object pose estimation. (Courtesy of [Xiang *et al.*, 2017])

ment, as shown in Fig. 21. Crivellaro et al. [Crivellaro *et al.*, 2018] predict the pose of each part of the object in the form of the 2D projections of a few control points with the assistance of a Convolutional Neural Network(CNN). Hu et al. [Hu *et al.*, 2018] proposed a segmentation-driven 6D pose estimation framework where each visible part of the object contributes to a local pose prediction in the form of 2D keypoint locations. The pose candidates are them combined into a robust set of 3D-to-2D correspondences from which the reliable pose estimation result is computed.

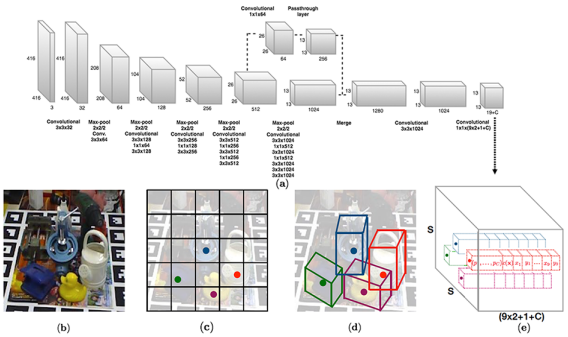

Figure 21: 6D object pose prediction of [Tekin *et al.*, 2018]. (a) The 6D object pose prediction CNN architecture. (b) An example input image with four objects. (c) The $S \times S$ grid showing cells responsible for detecting the four objects. (d) Each cell predicts 2D locations of the corners of the projected 3D bounding boxes in the image. (e) The 3D output tensor from our network, which represents for each cell a vector consisting of the 2D corner locations, the class probabilities and a confidence value associated with the prediction. (Courtesy of [Tekin *et al.*, 2018])

### Challenges

There exist three challenges for this kind of methods. The first challenge lies in that current methods show obvious limitations in cluttered scenes in which occlusions usually occur. The second one is the lack of sufficient training data, as the sizes of the datasets presented above are relatively small. Meanwhile, these methods still show poor performance on objects which do not exist in the training dataset.

## 3.4 Grasp Detection

Grasp detection is defined as being able to recognize the grasping points or the grasping pose for an object in any given images [Mahler *et al.*, 2017]. As stated in [Sahbani *et al.*, 2012], a grasping strategy should ensure stability, task compatibility and adaptability to novel objects, and the grasp quality can be measured with the location of contact points on the object and the hand configuration [Roa and Suárez, 2015]. There exist analytical approaches and empirical approaches in order to grasp a novel object and accomplish a following task. Analytical methods choose the finger positions and the hand configuration with kinematical and dynamical formulations of the grasp stability or the task requirements, and empirical methods use learning algorithms to choose a grasp that depend on the specific task and the target object's geometry. Based on whether or not the object localization, the object pose is required, we further divide them into three kinds, as shown in Table 3. Methods with known localization and pose will be discussed in this section. Methods with known localization and without pose, methods without localization and without pose, will be discussed in next two sections, respectively.

Table 3: Summary of grasp detection methods.

| Requirement | Methods | Key idea | Representative methods |
|---|---|---|---|
| With localization and pose | Empirical methods aiming at known objects | Utilize the pose and transform grasp points from know objects to the partial data | Zeng et al. [Zeng *et al.*, 2017], Billings and Roberson [Billings and Johnson-Roberson, 2018] |
| With localization and without pose | Analytical methods | Consider kinematics and dynamics formulation | Nguyen [Nguyen, 1987], Ponce et al. [Ponce *et al.*, 1993], Li et al. [Li *et al.*, 2003], Li and Sastry [Li and Sastry, 1988] |
| | Empirical methods aiming at familiar objects | Find mappings between familiar objects to the partial data | Miller et al. [Miller *et al.*, 2003a], Vahrenkamp et al. [Vahrenkamp *et al.*, 2016], Tian et al. [Tian *et al.*, 2018] |
| | Empirical methods aiming at unknown objects | Regress the grasp points by training CNN networks | Mahler et al. [Mahler *et al.*, 2017] |
| Without localization and without pose | End-to-end grasp detection | Inherit from object detection deep learning networks | Lenz et al. [Lenz *et al.*, 2015], Redmon and Angelova [Redmon and Angelova, 2015], Guo et al. [Guo *et al.*, 2016], Pinto and Gupta [Pinto and Gupta, 2016], Park et al. [Park *et al.*, 2018], Zeng et al. [Zeng *et al.*, 2018b] |

### Empirical methods aiming at known objects

Empirical methods or data-driven approaches [Bohg *et al.*, 2014; Caldera *et al.*, 2018] learn from previously known successful results produced with existing knowledge of grasping objects or simulations of robotic systems. According to [Bohg *et al.*, 2014], empirical methods can be divided depending on whether the target objects are know, familiar or novel objects. If the target object is known, it means that the 3D object and the grasp positions are known in the database. In that case, the 6D poses of the target object are estimated from the partial views, and they will become more accurate through fine tuning algorithms such as ICP. The grasp positions can be obtained directly from the complete 3D object.

This is the most popular method used for the grasping systems. Typical functional flow-chart of grasp detection aiming at known objects is illustrated in Fig. 22.

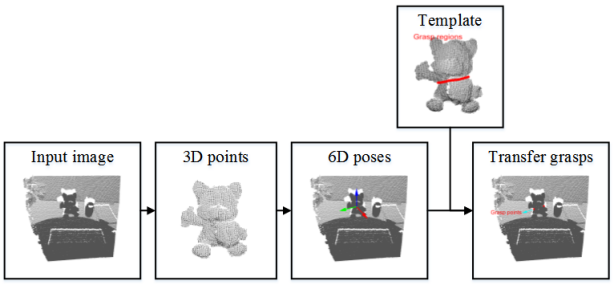


Figure 22: Typical functional flow-chart of grasp detection aiming at known objects.

Most of the methods [Zeng *et al.*, 2017] presented in the Amazon picking challenges utilized the 6D poses estimated through partial registration firstly. Zeng et al. [Zeng *et al.*, 2017] proposed an approach which segments and labels multiple views of a scene with a fully convolutional neural network, and then fits pre-scanned 3D object models to the segmentation results to obtain the 6D object poses, as shown in Fig. 23. Their method was part of the MIT-Princeton Team system that took 3rd- and 4th- place in the picking tasks at APC 2016. Besides, Billings and Johnson-Roberson [Billings and Johnson-Roberson, 2018] proposed a novel method which jointly accomplish object pose estimation and grasp point selection using a Convolutional Neural Network(CNN) pipeline. The pipeline takes in regions of interest proposals to simultaneously predict an intermediate silhouette representation which can regress the object pose. The grasp points are then generated from a precomputed database.

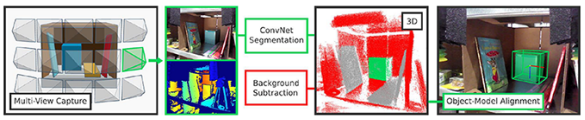


Figure 23: The robot captures color and depth images from 15 to 18 viewpoints of the scene. Each color image is fed into a fully convolutional network for 2D object segmentation. The result is integrated in 3D. The point cloud will then go through background removal and then aligned with a pre-scanned 3D model to obtain its 6D pose. (Courtesy of [Zeng *et al.*, 2017])

### Challenges

When the accurate 3D model is available, the object could be grasped by estimating the 6D poses, and the accuracy is usually high. However, when existing 3D models are different from the target one, the 6D poses will have a large deviation, and this will lead to the failure of grasp.

## 3.5 Grasp Detection Without Pose Estimation

In this section, grasping points are detected without pose estimation, as shown in Fig. 3. This means that the grasping points can only be estimated through analyzing the input data, or finding correspondences with existing grasps. Analytical methods, empirical methods aiming at similar objects and unknown objects, are discussed in this section.

### Analytical methods

Analytical methods consider kinematics and dynamics formulation in determining grasps [Sahbani *et al.*, 2012]. Force-closure and task compatibility are two main conditions in completing the grasping tasks. The force-closure is satisfied to afford a stable grasp, which only affords simple picking tasks. Whereas, task compatible grasps support specific tasks and optimally accomplish the grasping task. Typical functional flow-chart of analytical grasp detection methods is illustrated in Fig. 24.

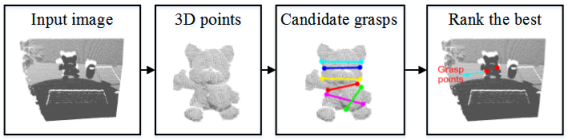


Figure 24: Typical functional flow-chart of analytical grasp detection methods.

There exist many force-closure grasp synthesis methods for 3D objects. Among them, the polyhedral objects are firstly dealt with, as they are composed of a finite number of flat faces. The force-closure condition is reduced into the test of the angles between the faces normals [Nguyen, 1987] or using the linear model to derive analytical formulation for grasp characterization [Ponce *et al.*, 1993; Liu, 1999; Ding *et al.*, 2000]. These methods did not consider the issue of selecting a grasping facet and conduct exhaustive searches. To handle the commonly used objects which usually have more complicated shapes, methods placing no restrictions on the object model by observing different contact points are proposed [Li *et al.*, 2003; Ding *et al.*, 2001; Liu *et al.*, 2004]. These methods try to find contact points on a 3D object surface to ensure force-closure and compute the optimal grasp by minimizing an objective energy function according to a predefined grasp quality criterion [Mirtich and Canny, 1994; Zhu and Wang, 2003]. However, searching the grasp solution space is a complex problem which is quite time-consuming. Some heuristical techniques were then proposed to reduce the search space by generating a set of grasp candidates according to a predefined procedure [Borst *et al.*, 2003; Fischer and Hirzinger, 1997], or by defining a set of rules to generate the starting positions [Miller *et al.*, 2003b]. These methods can only produce one successful grasp and are thus suitable for dealing with rather simple tasks such as picking and placing.

The choice of an optimal grasp is usually determined by the specific task to perform, and this leads to the research on task-oriented grasps. To do that, task-related grasp quality measures based on the capabilities to generate wrenches [Li and Sastry, 1988] or a linear matrix inequality formalism [Haschke *et al.*, 2005] have been proposed. Prats et al. [Prats *et al.*, 2007] proposed to use hand-preshapes during the early grasp planning stages where the task was considered. These methods still face the problem of precisely

representing the task and are computationally unaffordable. Meanwhile, these methods could neither be adapted for new tasks nor for new objects. To solve these problems, Song et al. [Song *et al.*, 2018] proposed a general approach for planning grasps on 3D objects based on hand-object geometric fitting. They built a contact score map on a 3D object's voxelization and applied this score map and a hand's kinematic parameters to find a set of target contacts on the object surface. Although this method works well for cases in which a complete 3D shape is provided in the image, it will fail when only partial view of the shape is available.

### Empirical methods aiming at similar objects

In most cases, the target objects are not totally the same with the objects in the existing database. If an object comes from a class that is involved in the database, it is regarded as a similar object. After the localization of the target object, correspondences-based methods can be utilized to transfer the grasp points from the similar full 3D object to the current partial view object. These methods learn the grasp by observing the object without estimating its 6D pose, since the current target object is not totally the same with the objects in the database. Typical functional flow-chart of empirical methods aiming at similar objects is illustrated in Fig. 25.

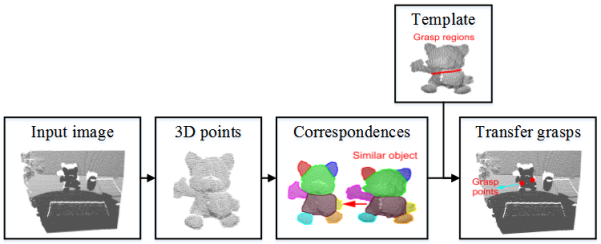


Figure 25: Typical functional flow-chart of empirical methods aiming at similar objects.

Different kinds of methods are utilized to find the correspondences based on taxonomy, segmentation, and so on. Andrew et al. [Miller *et al.*, 2003a] proposed a taxonomy-based approach, which classified objects into categories that should be grasped by each canonical grasp. Vahrenkamp et al. [Vahrenkamp *et al.*, 2016] presented a part-based grasp planning approach to generate grasps that are applicable to multiple familiar objects, as shown in Fig. 26. The object models are segmented according to their shape and volumetric information, and the objet parts are labeled with semantic and grasping information. A grasp transferability measure is proposed to evaluate how successful planned grasps are applied to novel object instances of the same object category. Tian et al. [Tian *et al.*, 2018] proposed a method to transfer grasp configurations from prior example objects to novel objets, which assumes that the novel and example objects have the same topology and similar shapes. They perform 3D segmentation on the objects considering geometric and semantic shape characteristics, compute a grasp space for each part of the example object using active learning, and build bijective contact mapping between the model parts and the corresponding grasps for novel objects.

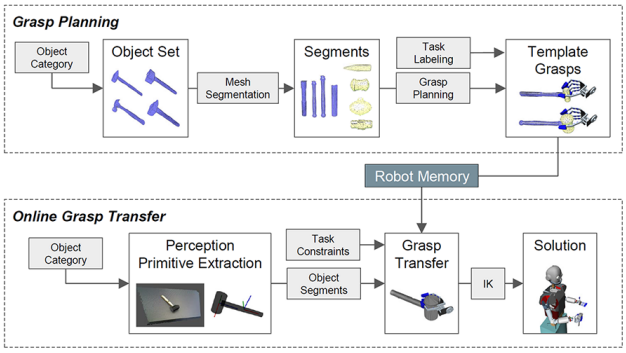


Figure 26: Part-based grasp planning is performed on multiple objects of an object category. The resulting grasping information is evaluated according to the expected transferability to novel objects. During online processing, grasping information is applied to novel objects which are segmented according to their RGB-D appearance. (Courtesy of [Vahrenkamp *et al.*, 2016])

### Empirical methods aiming at unknown objects

Above empirical methods of robotic grasping are performed based on the premise that certain prior knowledge, such as object geometry, physics models, or force analytic, are known. The grasp database usually covers a limited amount of objects, and empirical methods will face difficulties in dealing with unknown object. Whereas, the grasp experience learned before could provide the quality measurements when dealing with unknown objects. Typical functional flow-chart of empirical methods aiming at unknown objects is illustrated in Fig. 27. Mahler et al. [Mahler *et al.*, 2017] proposed a deep learning-based method to plan robust grasps with synthetic point clouds and analytic grasping metrics, as shown in Fig. 28. They first segment the current points of interests from the depth image, and multiple candidate grasp points are generated. The grasp qualities are then measured and the one with the highest quality will be selected as the final grasp point. Their database have more than 50k grasps, and the grasp quality measurement network achieved relatively satisfactory performance.

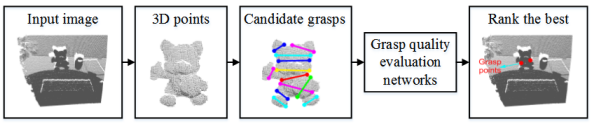


Figure 27: Typical functional flow-chart of empirical methods aiming at unknown objects.

### Challenges

This kind of methods rely on the accuracy of object segmentation. However, training a network which supports a wide range of objects is not easy. Meanwhile, these methods require the 3D object to grasp be similar enough to those of the annotated models such that correspondences can be found. It is also challenging to compute grasp points with high qualities for objects in cluttered environments where occlusion usually occurs.

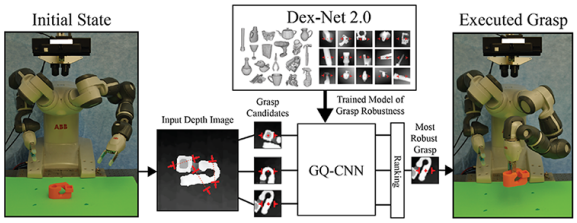


Figure 28: Dex-Net 2.0 Architecture. Left: The capture system with a depth camera. Middle: The Grasp Quality Convolutional Neural Network (GQ-CNN) is trained offline to predict the robustness candidate grasps from depth images. Several hundred grasp candidates are generated to the GQ-CNN network. Right: The most robust grasp candidate is determined and executed with the ABB YuMi robot. (Courtesy of [Mahler *et al.*, 2017])

### 3.6 End-to-end Grasp Detection

In this section, the grasp positions are detected in an end-to-end manner, which means the localization of the target object is skipped and the grasp positions are recovered directly from the input image, as shown in Fig. 3. Typical functional flow-chart of end-to-end grasp detection is illustrated in Fig. 29. This kind of methods can be divided into two-stage and one-stage methods. The former method first estimates candidate grasp positions and then selects the most likely one, while the latter method regress grasp positions directly.

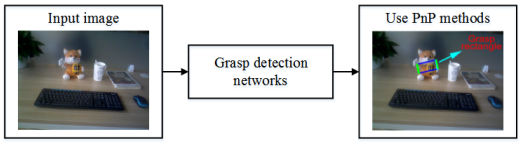


Figure 29: Typical functional flow-chart of end-to-end grasp detection.

In the two-stage method, the sliding window strategy is commonly used for detecting 2D robotic grasps. Lenz et al. [Lenz *et al.*, 2015] presented a two-step cascaded system with two deep networks, where the top detection results from the first are re-evaluated by the second, as shown in Fig. 30. The first network has fewer features, is faster to run, and can effectively prune out unlikely candidate grasps. The second, with more features, is slower but has to run only on the top few detections. Even though they achieved a high accuracy, the iterative scanning makes the process very slow. ten Pas et al. [ten Pas *et al.*, 2017] proposed a method for generating grasp hypotheses on any visible surface without requiring a precise segmentation of the target object. They also proposed a new grasp descriptor that incorporates surface normals and multiple views. However, multiple objects may be treated as a single atomic object since instance-level segmentation is not conducted.

Since a uniform network would perform better than the two-cascaded system [Lenz *et al.*, 2015], more and more one-stage methods have been proposed. Redmon and Angelova [Redmon and Angelova, 2015] proposed a larger neural network, which performs a single-stage regression to obtain graspable bounding boxes without using standard sliding

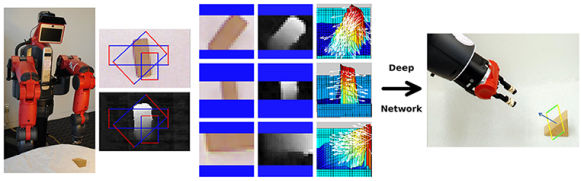


Figure 30: Detecting and executing grasps: From left to right: The system obtains an RGB-D image from a Kinect mounted on the robot, and searches over a large space of possible grasps, for which some candidates are shown. For each of these, it extracts a set of raw features corresponding to the color and depth images and surface normals, then uses these as inputs to a deep network which scores each rectangle. Finally, the top-ranked rectangle is selected and the corresponding grasp is executed using the parameters of the detected rectangle and the surface normal at its center. Red and green lines correspond to gripper plates, blue in RGB-D features indicates masked-out pixels. (Courtesy of [Lenz *et al.*, 2015])

window or region proposal techniques. Guo et al. [Guo *et al.*, 2016] presented a shared convolutional neural network to conduct object discovery and grasp detection. Pinto and Gupta [Pinto and Gupta, 2016] proposed a method to predict grasp locations via trial and error, as shown in Fig. 31. They trained a CNN-based classifier to estimate the grasp likelihood for different grasp directions given an input image patch. Chu et al. [Chu *et al.*, 2018] introduced a network composed of a grasp region proposal component and a robotic grasp detection component. Park et al. [Park *et al.*, 2018] proposed an accurate robotic grasp detection algorithm using fully convolutional neural networks with high-resolution images to recover the five poses $(x, y, \alpha, w, h)$ for manipulation. Zeng et al. [Zeng *et al.*, 2018b] presented a system capable of picking and recognizing novel objects with limited prior knowledge. This system first uses a category-agnostic affordable prediction algorithm to select among four different grasping primitive behaviors, and then recognizes the grasped objects by matching them to their product images. This method took 1st place in the stowing task of the Amazon Robotics Challenge 2017. Anyway, this method is not designed to pick a desired object for which the state of the target is needed.

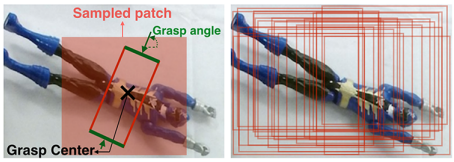


Figure 31: 1.5 times the gripper size image patch is utilized to predict the grasp-ability of a location and the angle is set at which it can be grasped. At test time, patches are sampled at different positions and the top graspable location and corresponding gripper angle are selected. (Courtesy of [Pinto and Gupta, 2016])

#### Challenges

For the end-to-end grasp detection methods, the computed grasp point may not be the globally optimal one, as only part of the object is available in the image. Meanwhile, they only

considered the geometric information, while other important factors like material and weight are missing.

## 3.7 Motion Planning

This section introduces the open-loop methods where the grasp points are assumed to have been detected with the above mentioned procedures. These methods design the path from the robot hand to the grasp points on the target object. Here motion representation is the key problem. Although there exist an infinite number of trajectories from the robotic hand to the target grasp points, many areas could not be reached due to the limitations of the robotic arm. Therefore, the trajectories need to be planned. There exist three kinds of methods in the literatures, which are traditional DMP-based methods, imitating learning-based methods, and reinforcement learning-based methods, as shown in 4. Typical functional flow-charts of DMP-based methods, imitating learning-based methods, and reinforcement learning-based methods to move to the grasp point, are illustrated in Fig. 32.

Table 4: Summary of motion planning methods.

| Requirement | Methods | Key idea | Representative methods |
|---|---|---|---|
| With grasp point | DMP-based methods | Formalized as stable nonlinear attractor systems | DMP [Schaal, 2006], Rai et al. [Rai *et al.*, 2017] |
| | Imitating learning | Learning from demonstration | Amor et al. [Amor *et al.*, 2012] |
| | Reinforcement learning to move to the grasp point | Self-supervised learning | Kwiatkowski and Lipson [Kwiatkowski and Lipson, 2019] |
| Without grasp point | Reinforcement learning to grasp the object | Treat successful grasps as the reward function | Levine et al. [Levine *et al.*, 2018], Kalashnikov et al. [Kalashnikov *et al.*, 2018], Frederik et al. [Ebert *et al.*, 2018] |

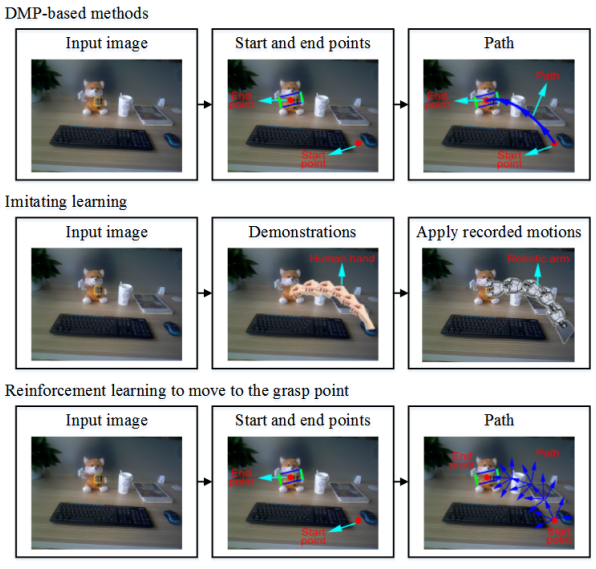


Figure 32: Typical functional flow-chart of motion planning methods.

### Traditional methods

Traditional methods consider the dynamics of the motion and generate motion primitives. The Dynamic Movement Primitives (DMPs) [Schaal, 2006] are one of the most popular motion representations that can serve as an reactive feedback controller. DMPs are units of action that are formalized as stable nonlinear attractor systems. They encode kinematic control policies as differential equations with the goal as the attractor [Rai *et al.*, 2017]. A nonlinear forcing term allows shaping the transient behavior to the attractor without endangering the well-defined attractor properties. Once this nonlinear term has been initialized, e.g., via imitation learning, this movement representation can be generalized with respect to task parameters such as start, goal, and duration of the movement.

DMPs have been successfully used to in imitation learning, reinforcement learning, movement recognition and so on. Colomé et al. [Colomé and Torras, 2018] addressed the process of simultaneously learning a DMP-characterized robot motion and its underlying joint couplings through linear dimensionality reduction, which provides valuable qualitative information leading to a reduced and intuitive algebraic description of such motion. Rai et al. [Rai *et al.*, 2017] proposed learning feedback terms for reactive planning and control. They investigate how to use machine learning to add reactivity to a previously learned nominal skilled behaviors represented by DMPs. Pervez and Lee [Pervez and Lee, 2017] proposed a generative model for modeling the forcing terms in a task parameterized DMP. Li et al. [Li *et al.*, 2018a] proposed an enhanced teaching interface for a robot using DMP and Gaussian Mixture Model (GMM). The movements are collected from a human demonstrator by using a Kinect v2 sensor. GMM is employed for the calculation of the DMPs, which model and generalize the movements.

### Imitation learning

This kind of method is also known as learning from demonstration. For the traditional methods, the kinematics of the robot is ignored, as it assumes that any set of contacts on the surface of the object can be reached, and arbitrary forces can be exerted at those contact points. Anyway, the actual set of contacts that can be made by a robotic hand is severely limited by the geometry of the hand. Through imitation learning, the grasp actions learned from successful grasps could be mapped to the grasping of target object in a more natural way. The movements from the demonstration can be decomposed into DMPs. When grasping the same or similar object, the same movement trajectory can be utilized.

If the target object exists in a stored database, the grasp points could be directly obtained. The problem then becomes finding a path from the start-point to the end-point to reach the target object, and grasping the object with particular poses. If the target object is similar with example objects, the grasp point of the target object can be obtained through the methods in Section 3.5. The target object will be compared with those in the database and a demonstrated grasp used on a similar object will be selected. The target object can be considered as a transformed version of the demonstration object, and the grasp points can be mapped from one

object to the other. Amor et al. [Amor *et al.*, 2012] presented an imitation learning approach for learning and generalizing grasping skills based on human demonstration, as shown in Fig. 33. They split the task of synthesizing a grasping motion into three parts: learning efficient grasp representations from human demonstrations, warping contact points onto new objects, and optimizing and executing the reach-and-grasp movements. Their method can be used to easily program new grasp types into a robot and the user only needs to perform a set of example grasps.

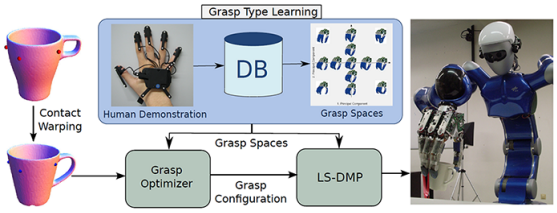


Figure 33: The contact points of a known object are warped on the current object. Using the resulting positions, an optimizer finds the ideal configuration of the hand during the grasp. The optimizer uses low-dimensional grasp spaces learned from human demonstrations. Finally, a latent space dynamic motor primitive robustly executes the optimized reach-and-grasp motion. (Courtesy of [Amor *et al.*, 2012])

### Reinforcement learning to move to the grasp point

There exist methods utilizing reinforcement learning to achieve more natural movements [Kwiatkowski and Lipson, 2019]. Aiming at the low control resolution of the robotic arm, Kwiatkowski and Lipson [Kwiatkowski and Lipson, 2019] proposed a self-modeling machine for the robotic arm, as shown in Fig. 34. They build the trajectory space of the robotic arm through random sampling, and then plan the path to the grasp position. This self-model can be applied to different tasks such as pick-and-place and handwriting. Their method provides a useful way to use simulators.

### Methods considering obstacles

Sometimes, the robot cannot approach the target object for reasons like constrained space, various obstacles and so on. Meanwhile, the obstacles may be too large for the robot to grasp. This requires the robot's interaction with the environment. The most commonly seen solution for such grasping tasks is the object-centric method [Laskey *et al.*, 2016], which separates the target and the environment. This kind of method works well in structured or semi-structured settings where objects are well-separated. There also exists a clutter-centric approach [Dogar *et al.*, 2012], which utilizes action primitives to make simultaneous contact with multiple objects. With this approach, the robot reaches for and grasps the target while simultaneously contacting and moving aside objects to clear a desired path.

### Challenges

The main challenge of this kind of methods is that it heavily depends on the accuracy of grasp detection. If the grasp positions are detected accurately, the motion planing will achieve a high rate of success. The efficiency of escaping the obstacles is also a challenge for practical robot operations.

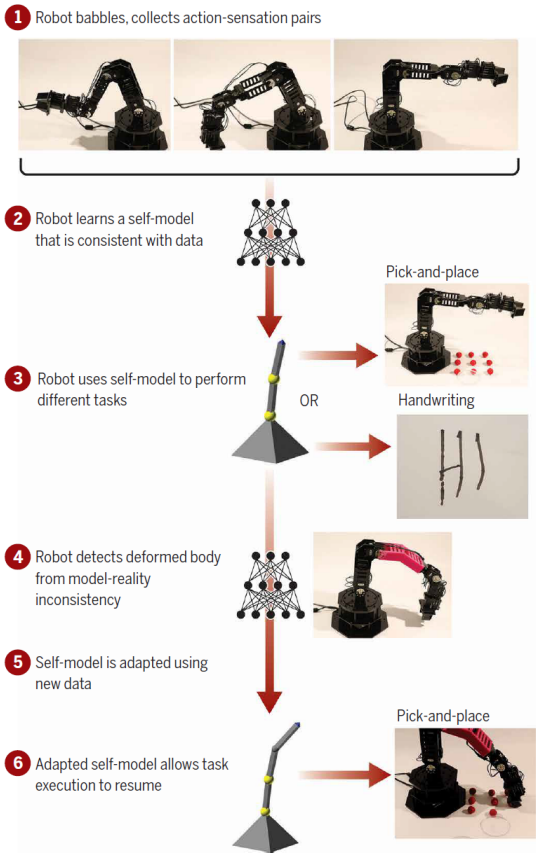


Figure 34: Self-model generation, usage, and adaptation. An outline of the self-modeling process from data collection to task planning. (Step 1) The robot recorded action-sensation pairs. (Step 2) The robot used deep learning to create a self-model consistent with the data. (Step 3) The self-model could be used for internal planning of two separate tasks without any further physical experimentation. (Step 4) The robot morphology was abruptly changed to emulate damage. (Step 5) The robot adapted the self-model using new data. (Step 6) Task execution resumed. (Courtesy of [Kwiatkowski and Lipson, 2019])

## 3.8 End-to-end Motion Planning

This section introduces the close-loop methods where the grasp points are not given, as shown in Fig. 3. And typical functional flow-chart of end-to-end motion planning is illustrated in Fig. 35. These methods directly accomplish the grasp task after given an original RGB-D image by using reinforcement learning.

### Reinforcement learning to grasp the object

The reward function is defined relating to the state of grasping. Viereck et al. [Viereck *et al.*, 2017] proposed an approach to learn a closed-loop controller for robotic grasping that dynamically guides the gripper to the object. Levine et al. [Levine *et al.*, 2018] proposed a learning-based method for hand-eye coordination in robotic grasping from monocular

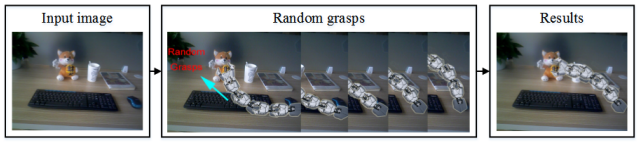


Figure 35: Typical functional flow-chart of end-to-end motion planning.

images. They utilized the grasp attempts as the reward function, and trained a large convolutional neural network to predict the probability that task-space motion of the gripper will result in successful grasps. The architecture of the CNN grasp predictor is shown in Fig. 36. Kalashnikov et al. [Kalashnikov *et al.*, 2018] utilized QT-Opt, a scalable self-supervised vision-based reinforcement learning framework, to perform closed-loop real-world grasping. Their method shows great scalability to unseen objects. Besides, it is able to automatically learn regrasping strategies, probe objects to find the most effective grasps, learn to reposition objects and perform other non-prehensile pre-grasp manipulations, and respond dynamically to disturbances and perturbations. Frederik et al. [Ebert *et al.*, 2018] proposed a method on learning robotic skills from raw image observations by using autonomously collected experience. They devise a self-supervised algorithm for learning image registration to keep track of the objects of interest. Experimental results demonstrate that unlabeled data is successfully used to perform complex manipulation tasks. Fang et al. [Fang *et al.*, 2018] proposed a task-oriented grasping network, which is a learning-based model for jointly predicting task-oriented grasps and subsequent manipulation actions given the visual inputs. Besides, they employed the self-supervised learning paradigm to allow the robot perform grasping and manipulation attempts with the training labels automatically generated.

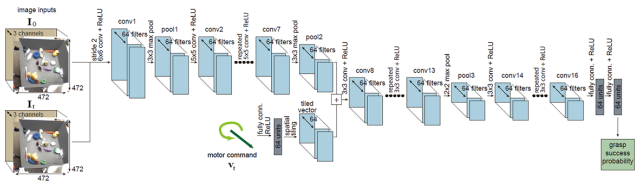


Figure 36: The architecture of the CNN grasp predictor [Levine *et al.*, 2018]. The input image $I_t$, as well as the pregrasp image $I_0$, are fed into a 6 $\times$ 6 convolution with stride 2, followed by 3$\times$3 max-pooling and 6 5$\times$5 convolutions. This is followed by a 3$\times$3 max-pooling layer. The motor command $v_t$ is processed by one fully connected layer, which is then pointwise added to each point in the response map of pool2 by tiling the output over the special dimensions. The result is then processed by 6 3 $\times$ 3 convolutions, 2 $\times$ 2 max-pooling, 3 more 3 $\times$ 3 convolutions, and two fully connected layers with 64 units, after which the network outputs the probability of a successful grasp through a sigmoid. Each convolution is followed by batch normalization. (Courtesy of [Levine *et al.*, 2018])

### Methods considering obstacles

The above mentioned methods assume that no obstacles exist on the path between the robotic hand and the grasp positions on target object. There also exist some methods which learn to push the obstacles away and grasp the target object in a closed-loop manner. Zeng et al. [Zeng *et al.*, 2018a] proposed a model-free deep reinforcement learning method to discover and learn the synergies between pushing and grasping. Their method involves two convolutional networks that map from visual observations to actions. The two networks are trained jointly in a Q-learning framework and are entirely self-supervised by trial and error, where rewards are provided from successful grasps. With picking experiments in both simulation and real-word scenarios, their system quickly learns complex behaviors aiming challenging clutter cases and achieves better grasping success rates and picking efficiencies.

### Challenges

The first challenge lies in that the data generated by simulation show limited efficiency, although there have been domain adaption methods [Bousmalis *et al.*, 2018a] proposed to improve efficiency of deep robotic grasping. Meanwhile, it remains challenging on avoiding obstacles when trying to grasp the object. Especially when the structure of the robot arm should be considered, it is difficult to define the mathematical grasp model according to different tasks.

## 4 Datasets, Evaluations and Comparisons

Data plays a pivotal role in computer vision related research. In this section, the datasets, together with evaluations and comparisons with start-of-the-art methods, are presented for object detection, object segmentation, 6D pose estimation, grasp detection and grasp planning.

### 4.1 Object detection

#### Datasets

Datasets about 2D object detection are summarized comprehensively in [Liu *et al.*, 2018], and here we only present some representative datasets, such as PASCAL VOC [Everingham *et al.*, 2015], SUN [Xiao *et al.*, 2016], ImageNet [Russakovsky *et al.*, 2015], MS COCO [Lin *et al.*, 2014], Places [Zhou *et al.*, 2018] and Open Images [Krasin *et al.*, 2016]. These datasets are shown in Table 5. These datasets help to train the detection networks to provide the bounding box information, which can be used for computing the precise contours of the target object.

#### Evaluation metrics

Evaluation metrics about 2D object detection are also summarized in Liu et al. [Liu *et al.*, 2018], and here we briefly introduced the mean Average Precision (mAP) [Everingham *et al.*, 2010] metric.

The $AveragePrecision$ (AP) metric is derived from precision and recall, and is usually used to evaluate a specific category. The $meanAveragePrecision$ (mAP) [Everingham *et al.*, 2010] is averaged over all object categories, which is adopted as the measure of the general performance. The standard outputs of a detector applied to a testing image $I$ are the predicted detections $\{(b_j, c_j, p_j)\}_j$, indexed by $j$. Here $b$ denotes the predicted location such as the bounding box, $c$ denotes the category label and the confidence level is $p$. A

Table 5: Comparison between various publicly available 2D object detection datasets

| Dataset | Total images | Categories | Images size |
|---|---|---|---|
| PASCAL VOC [Everingham *et al.*, 2015] | 11540 | 20 | 470×380 |
| SUN [Xiao *et al.*, 2016] | 131,072 | 908 | 500×300 |
| ImageNet [Russakovsky *et al.*, 2015] | 14 millions+ | 21841 | 500×400 |
| MS COCO [Lin *et al.*, 2014] | 328,000+ | 91 | 640×480 |
| Places [Zhou *et al.*, 2018] | 10 millions+ | 434 | 256×256 |
| Open Images [Krasin *et al.*, 2016] | 9 millions+ | 6000+ | varied |

predicted detection $(b, c, p)$ is regarded as a True Positive(TP) if the predicted class label $c$ is the same as the ground truth label $c_g$, and the overlap ratio IOU(Intersection Over Union) between the predicted bounding box $b$ and the ground truth $b_g$ is not smaller than a predefined threshold $\varepsilon$. The definition of IOU is as follows:

$$IOU(b, b_g) = \frac{area(b \cap b_g)}{area(b \cup b_g)}. \quad (1)$$

Here $area(b \cap b_g)$ denotes the intersection of the predicted and ground truth bounding boxes, and $area(b \cup b_g)$ represents their union. The typical value of $\varepsilon$ is 0.5. If IOU is smaller than $\varepsilon$, it is considered as False Positive(FP). The confidence level $p$ is usually compared with a threshold $\beta$ to determine whether the predicted class label $c$ is accepted. Each detection is either a TP or FP, and the precision $P(\beta)$ with confidence threshold $\beta$ can be computed. The Average Precision (AP) can thus be achieved by varying $\beta$.

**Comparisons**

Comparisons of mAP of some typical object detection networks on COCO dataset are presented in Table 6.

Table 6: Detection mAP of object detection methods on MS COCO dataset.

| Method | IoU[0.5 0.95] | IoU0.5 |
|---|---|---|
| Fast RCNN(VGG16) | 0.2 | 0.36 |
| Faster RCNN(VGG16) | 0.22 | 0.43 |
| Faster RCNN(ResNet101) | 0.37 | 0.59 |
| Mask RCNN(ResNeXt+FPN) | 0.50 | 0.72 |

## 4.2 Object segmentation

A comprehensive reviews about object segmentation can be found at [Garcia-Garcia *et al.*, 2017], and here we focus on the representative 2D datasets and methods.

**Datasets**

Some generic 2D object segmentation datasets can be found below.

Table 7: Comparison between various publicly available generic 2D object segmentation datasets.

| Dataset | Total images | Categories | Images size |
|---|---|---|---|
| PASCAL VOC 2012 Segmentation [Everingham *et al.*, 2015] | 2913 | 21 | Variable |
| PASCAL-Context [Mottaghi *et al.*, 2014] | 19,740 | 540(59) | Variable |
| PASCAL-Part [Chen *et al.*, 2014] | 19,740 | 20 | Variable |
| SBD [Hariharan *et al.*, 2011] | 11,355 | 21 | Variable |
| MS COCO [Lin *et al.*, 2014] | 204,721 | 80+ | Variable |
| DAVIS [Perazzi *et al.*, 2016; Pont-Tuset *et al.*, 2017] | 8422 | 4 | 480p |

**Evaluation metrics**

Similar with object detection, object segmentation uses mean Intersection over Union(mIoU) as the evaluation metric. mIoU computes the ratio between the intersection and the union of the ground truth and the predicted segmentation areas. The ratio can be reformulated as the number of true positives over the sum of true positives, false negatives, and false positives. The IoU is computed on a per-class basis and then averaged as follows:

$$MIoU = \frac{1}{k+1} \sum_{i=0}^{k} \frac{p_{ii}}{\sum_{j=0}^{k} p_{ij} + \sum_{j=0}^{k} p_{ji} - p_{ii}} \quad (2)$$

**Comparisons**

Here we simply give the performance results of some typical segmentation networks on the PASCAL VOC-2012 dataset in Table 8. Among all these methods, DeepLab [Chen *et al.*, 2018] has the best performance.

Table 8: Performance results on PASCAL VOC-2012.

| Method | Accuracy(IoU) |
|---|---|
| DeepLab [Chen *et al.*, 2018] | 79.70 |
| Dilation [Yu and Koltun, 2015] | 75.30 |
| CRFasRNN [Zheng *et al.*, 2015] | 74.70 |
| ParseNet [Liu *et al.*, 2015] | 69.80 |
| FCN-8s [Long *et al.*, 2015b] | 67.20 |
| Bayesian SegNet [Kendall *et al.*, 2015] | 60.50 |

## 4.3 6D pose estimation

**Datasets**

There exist various benchmarks for 6D pose estimation. Hodaň et al. [Hodaň *et al.*, 2018b] proposed a benchmark for 6D pose estimation of a rigid object from a single RGB-D input image. They introduced the existing datasets such as LM/LM-O dataset [Hinterstoisser *et al.*, 2012], IC-MI/IC-BIN dataset [Tejani *et al.*, 2014], T-LESS dataset [Hodaň *et al.*, 2017], RU-APC dataset [Rennie *et al.*, 2015] and the TUD-L/TYO-L dataset [Hodaň *et al.*, 2018b]. Besides,

there also exist some other datasets such as the YCB-Video dataset [Xiang *et al.*, 2017], etc. Here we only reviewed some of the widely used datasets in Table 9 and their volumes are presented.

Table 9: Comparison between various publicly available 6D pose estimation datasets

| Dataset | Objects | Total images |
|---|---|---|
| LineMod [Hinterstoisser *et al.*, 2012] | 15 | 1100+ frame video sequences |
| T-LESS [Hodaň *et al.*, 2017] | 30 | 49K images |
| RU-APC [Rennie *et al.*, 2015] | 24 | 10000 images |
| YCB-Video [Xiang *et al.*, 2017] | 21 | 92 RGB-D videos |

## Evaluation metrics

In order to estimate the 6D pose of an object in the image, the pose error is measured. The object pose can be represented by a $4 \times 4$ matrix $\mathrm{P} = [R, t; 0, 1]$, where $R$ is a $3 \times 3$ rotation matrix and $t$ is a $3 \times 1$ translation vector. Anyway, direct comparison of the variances between rotations can not provide an intuitive visual understanding. Therefore, other evaluation metrics are proposed.

A commonly used metric is the Average Distance of Model Points(ADD) [Hinterstoisser *et al.*, 2012] for non-symmetric objects and the average closest point distances(ADD-S) [Xiang *et al.*, 2017] for symmetric objects. Given a 3D model $M$, the ground truth rotation $R$ and translation $T$, and the estimated rotation $\hat{R}$ and translation $\hat{T}$, we compute the average distance of all model points $x$ from their transformed versions.

$$e_{ADD} = \underset{x \in M}{avg} \left\| (Rx + T) - (\hat{R}x + \hat{T}) \right\|. \tag{3}$$

ADD-S [Xiang *et al.*, 2017] is an ambiguity-invariant pose error metric which takes both symmetric and non-symmetric objects into an overall evaluation. Given the estimated pose $[\hat{\mathrm{R}}|\hat{\mathrm{T}}]$ and ground truth pose $[\mathrm{R}|\mathrm{T}]$, ADD-S calculates the mean distance from each 3D model point transformed by $[\hat{\mathrm{R}}|\hat{\mathrm{T}}]$ to its closest neighbour on the target model transformed by $[\mathrm{R}|\mathrm{T}]$.

The visible surface discrepancy(VSD) [Hodaň *et al.*, 2018b] is another metric which measures the two rendered distance maps $\hat{S}$ and $\bar{S}$ of the object model according to the estimated pose $\hat{P}$ w.r.t. the ground truth pose $\bar{P}$. Given the visibility masks which contain the sets of pixels where the model $M$ is visible in the image $I$, the VSD error is calculated according to the two distance maps as:

$$\begin{aligned} & e_{VSD}(\hat{S}, \bar{S}, S_I, \hat{V}, \bar{V}, \tau) \\ & = \underset{p \in \hat{V} \cup \bar{V}}{avg} \begin{cases} 0 & if p \in \hat{V} \cap \bar{V} \wedge |\hat{S}(p) - \bar{S}(p)| < \tau \\ 1 & otherwise. \end{cases} \end{aligned} \tag{4}$$

where $\tau$ is a misalignment tolerance. The pose error $e_{VSD}$ is calculated over the visible part of the model surface and thus the indistinguishable poses are treated as equivalent.

## Comparisons

The accuracy of different methods is the core element to be measured, and other properties such as speed and robustness to occlusions are also introduced here.

Hodaň et al. [Hodaň *et al.*, 2018b] utilized the VSD metric, where the misalignment tolerance $\tau$ is set to 20 mm and the correctness threshold $\theta$ is set to 0.3. They compared many algorithms and the representative methods are presented in Table 10. Among them, Buch-16 [Buch *et al.*, 2016] is the correspondence-based method, Hodaň [Hodaň *et al.*, 2015] is the template-based method, and Drost-10 [Drost *et al.*, 2010], Brachmann-14 [Brachmann *et al.*, 2014] and Vidal-18 [Vidal *et al.*, 2018] are voting-based methods. It was concluded that method of Vidal-18 [Vidal *et al.*, 2018] which is based on the point-pair features outperformed the other methods. However, comparisons with regression-based methods were not conducted.

Table 10: Accuracies of various methods using VSD metric

| Category | Method | Average | Time(s) |
|---|---|---|---|
| Correspondence-based methods | Buch-16 [Buch *et al.*, 2016] | 7.20 | 47.1 |
| Template-based methods | Hodaň-15 [Hodaň *et al.*, 2015] | 67.23 | 13.5 |
| Voting-based methods | Drost-10 [Drost *et al.*, 2010] | 68.06 | 2.3 |
| | Brachmann-14 [Brachmann *et al.*, 2014] | 34.61 | 1.4 |
| | Vidal-18 [Vidal *et al.*, 2018] | 74.60 | 4.7 |

Wang et al. [Wang *et al.*, 2019a] proposed a generic framework for estimating the 6D poses of a set of known objects from RGB-D images and conducted comparisons with regression-based methods using the ADD-S metric on YCB-video dataset. The results are shown in Table 11. Besides, they also conducted experiments on the LineMOD dataset using the ADD metric, and the results are shown in Table 12. From the tables, we can see that DenseFusion achieved the highest accuracy comparing with other regression-based methods. This is because DenseFusion employed a novel local feature fusion scheme using both RGB and depth images, whereas other methods only utilized single source data.

Table 11: Accuracies of regression-based methods using ADD-S metric on YCB-video dataset.

| Method | AUC | <2cm |
|---|---|---|
| PointFusion [Xu *et al.*, 2018] | 83.9 | 74.1 |
| PointCNN+ICP [Xiang *et al.*, 2017] | 93.0 | 93.2 |
| DenseFusion [Xiang *et al.*, 2017](iterative) | 93.1 | 96.8 |

Table 12: Accuracies of regression-based methods using ADD metric on LineMOD dataset.

| Category | Method | Average |
|---|---|---|
| RGB-based methods | BB8 [Rad and Lepetit, 2017] | 62.7 |
| | SSD-6D [Kehl *et al.*, 2017] | 79 |
| | Tekin et al. [Tekin *et al.*, 2018] | 55.95 |
| | PoseCNN [Xiang *et al.*, 2017] | 62.7 |
| | PoseCNN+DeepIM [Xiang *et al.*, 2017; Li *et al.*, 2018c] | 88.6 |
| RGB-D-based methods | Implicit [Sundermeyer *et al.*, 2018]+ICP | 64.7 |
| | SSD-6D [Kehl *et al.*, 2017]+ICP | 79 |
| | PointFusion [Xu *et al.*, 2018] | 73.7 |
| | DenseFusion [Wang *et al.*, 2019a](per-pixel) | 86.2 |
| | DenseFusion [Wang *et al.*, 2019a](iterative) | 94.3 |

## 4.4 Grasp detection

### Datasets

Although there exist a few datasets in which the real data is used, most datasets for grasp detection utilized simulation data to compensate for the lack of real grasping data. Some important datasets are presented in Table 13.

Table 13: Comparison between various publicly available robotic grasp datasets.

| Dataset | Objects Num | RGB image num | Depth image num |
|---|---|---|---|
| Stanford Grasping | 10 | 13747 | 13747 |
| Cornell Grasping | 240 | 885 | 885 |
| YCB Benchmarks | 77 | 46200 | 46200 |
| CMU dataset | over 150 | 50567 | no |
| Google dataset | not mentioned | 800000 | no |
| Dex-Net 1.0 | over 150 | 50567 | no |
| Dex-Net 2.0 | over 150 | 50567 | no |
| JACQUARD | 11619 | 54485 | 108970 |

### Evaluation metrics

There exist two representations for the grasp configuration: the point defined grasps and the rectangle defined grasps [Jiang *et al.*, 2011]. The Point defined grasp only suggest where to grasp an object and do not determine the width and orientation of the gripper ends. To overcome its limitation, Jiang et al. [Jiang *et al.*, 2011] represented a grasp as an oriented rectangle, where the upper-left corner, length, width and its angle from the x-axis are included. Redmon et al. [Redmon and Angelova, 2015] then updated the representation of a grasp rectangle, where the center of the rectangle, height, width and the orientation of the rectangle relative to the horizontal axis are used.

There exist two metrics for evaluating the performance of grasp detection: the point metric and the rectangle metric. The former evaluates the distance between predicted grasp center and the actual grasp center w.r.t. a threshold value. It has difficulties in determining the distance threshold and does not consider the grasp angle. The latter metric considers a grasp to be correct if the grasp angle is within $30^\circ$ of the ground truth grasp, and the Jaccard index $J(A, B) = |A \cap B|/|A \cup B|$ of the predicted grasp $A$ and the ground truth $B$ is greater than $25\%$.

Besides the metric on comparing the estimated grasp configuration with the ground truth, there also exist other metrics [Mahler *et al.*, 2017] to evaluate the performance of the predicted grasp points, including:

1) Success rate or accuracy: The percentage of grasps that were able to lift, transport, and hold a desired object after shaking.
2) Precision: The success rate on grasps that are have an estimated robustness higher than $50\%$.
3) Robust grasp rate: The percentage of planned grasps with an estimated robustness high than $50\%$.
4) Planning time: The time in seconds between receiving an image and returning a planned grasp.

### Comparisons

The performance of various grasp detection methods on DexNet2.0 [Mahler *et al.*, 2017] are listed in Table 14. In this table, IGQ represents the image-based grasp quality metric, where a set of force closure grasp candidates are firstly sampled [Chen and Burdick, 1993] and ranked. REG, ML-RF, ML-SVM and GQ-L-Adv represent the point cloud registration method [Hernandez *et al.*, 2016], Random Forest-based machine learning method, Support Vector Machine-based machine learning method, and the Grasp Quality CNN model-based method, respectively.

Table 14: Comparison of grasping methods using different grasp points.

| Method | Success Rate (%) | Precision (%) | Robust Grasp Rate (%) | Planning Time ($sec$) |
|---|---|---|---|---|
| IGQ | $70 \pm 10$ | N/A | N/A | 1.9 |
| ML-RF | $75 \pm 9$ | 100 | 5 | 0.8 |
| ML-SVM | $80 \pm 9$ | 100 | 0 | 0.9 |
| REG | $95 \pm 5$ | N/A | N/A | 2.6 |
| GQ-L-Adv | $93 \pm 6$ | 94 | 43 | 0.8 |

## 4.5 Grasp planning

The grasp planning datasets could be generated from either manipulations of real robotic arms or simulated environments. The robot grasps the object randomly, and successful grasps are recorded. The evaluation of grasp planning is usually conducted by computing the success rate of a grasp attempt. As the methods differ in the setup, target objects, grippers and clutter, we only list the datasets in Table 15 without comparisons.

Manipulations of real robotic arms is straightforward but cost expensive. Pinto and Gupta [Pinto and Gupta, 2016] proposed a dataset size of 50K data points collected over 700 hours of robotic grasping attempts. This allows them to train a Convolutional Neural Network (CNN) to predict grasp locations without severe overfitting. Levine et al. [Levine *et al.*, 2018] collected over 800,000 grasp attempts over two

Table 15: Grasp planning datasets.

| Category | Dataset | Information |
|---|---|---|
| Real grasp datasets | Pinto and Gupta [Pinto and Gupta, 2016] | 50K grasps |
| | Levine et al. [Levine *et al.*, 2018] | 800,000 grasps |
| | Wang et al. [Wang *et al.*, 2019b] | 2550 sets |
| Grasp simulators | Andrew and Peter [Miller and Allen, 2004] | GraspIt! |
| | León et al. [León *et al.*, 2010] | OpenGRASP |
| | Quillen et al. [Quillen *et al.*, 2018] | Simulated grasping |

months, using robotic manipulators between 6 and 14. Wang et al. [Wang *et al.*, 2019b] introduced a visual-tactile multimodal grasp dataset built by a designed dexterous robot hand – the Intel's Eagle Shoal robot hand. The dataset contains 2550 data volumes, including tactile, joint, time label, image, and RGB and depth sequences.

Various grasp simulation toolkits are developed to facilitate the grasps generation. Andrew and Peter [Miller and Allen, 2004] proposed GraspIt! – a versatile simulator for robotic grasping. GraspIt! supports the loading of objects and obstacles of arbitrary geometry to populate a complete simulation world. It allows a user to interactively manipulate a robot or an object and create contacts between them. Xue et al. [Xue *et al.*, 2009] implemented a grasping planning system based on GraspIt! to plan high-quality grasps. León et al. [León *et al.*, 2010] presented OpenGRASP, a toolkit for simulating grasping and dexterous manipulation. It provides a holistic environment that can deal with a variety of factors associated with robotic grasping. Quillen et al. [Quillen *et al.*, 2018] proposed a simulated grasping benchmark for a robotic arm with a two-finger parallel jaw gripper on grasping random objects from a bin. They also present an empirical evaluation of off-policy deep reinforcement learning algorithms on vision-based robotic grasping tasks.

## 5 Challenges and Future Directions

In this survey, we reviewed related works on robotic grasping from four key aspects: object detection, 6D pose estimation, grasp detection and grasp planning. The goal is to allow the readers get a comprehensive map about how to conduct a successful grasp given the initial raw data. Various methods are introduced in each part, as well as the datasets and the comparisons. Comparing with existing literatures, we present an end-to-end review about how to conduct robotic grasping.

Traditional grasping methods work well on known objects, while it is difficult for them to extend the current grasp ability to novel objects. Besides, the incomplete data caused by the interaction of different objects in cluttered environment also proposed a big challenge to the grasping task. To handle these problems, the deep learning-based methods are proposed. These methods show better generalization abilities to novel objects, and are good at handling objects in cluttered environment. However, they still rely on a huge number of labeled training data.

The future directions to assist in robotic grasping include four aspects: data acquisition, semantic perception, grasp decision and data training.

Data acquisition is closely related to the performance of robotic grasping. Firstly, the sensor error can be reduced. The accuracy of the optical sensors is limited comparing to the laser devices, and the sensor error is introduced. Whereas, it can be reduced through reinforcement learning to meet the requirement of robotic grasping; Secondly, more accurate segmentation methods can be employed to handle the partial view data which is usually seen in the captured images; Thirdly, a more widely perspective data can be utilized as the partial views are not enough to get a comprehensive knowledge of the target object. To do that, methods on using the poses or robotic arms or the RGB-D slam methods can be adopted to merge the multi-view data.

Semantic perception means obtaining the high-level information of the input data. Firstly, with the help of various segmentation methods, parts of the object instead of the complete shape can be used to decrease the candidate grasping points. Secondly, the surface material and the weight information can be estimated to obtain more precise grasping detection results. These information can be regressed through learning-based methods.

While the grasping decisions are made in an open-loop manner in most of the current methods, they should be carried out in a closed-loop manner, for which feedbacks are given from previous grasps rather than one-shot estimation. Besides, the grasp for specific tasks should be computed with task-specific learning approaches instead of general solutions.

The training data plays a pivotal role for deep learning-based methods. Although the simulation methods can help to enrich the data, there still exists gaps from the simulation data to the practical one. To alleviate this problem, many domain adaptation methods [Bousmalis *et al.*, 2018b] have been proposed. Besides, the semi-supervised learning approaches can also be utilized to learn to grasp with incorporate unlabeled data.

## References


[Ali *et al.*, 2018] Waleed Ali, Sherif Abdelkarim, Mahmoud Zidan, Mohamed Zahran, and Ahmad El Sallab. Yolo3d: End-to-end real-time 3d oriented object bounding box detection from lidar point cloud. In *European Conference on Computer Vision*, pages 716–728. Springer, 2018.

[Amor *et al.*, 2012] Heni Ben Amor, Oliver Kroemer, Ulrich Hillenbrand, Gerhard Neumann, and Jan Peters. Generalization of human grasping for multi-fingered robot hands. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2043–2050. IEEE, 2012.

[Badrinarayanan *et al.*, 2017] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[Bay *et al.*, 2006] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[Besl and McKay, 1992a] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*, volume 1611, pages 586–607. International Society for Optics and Photonics, 1992.

[Besl and McKay, 1992b] Paul J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256, February 1992.

[Bicchi and Kumar, 2000] Antonio Bicchi and Vijay Kumar. Robotic grasping and contact: A review. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 348–353. IEEE, 2000.

[Billings and Johnson-Roberson, 2018] Gideon Billings and Matthew Johnson-Roberson. Silhonet: An RGB method for 3d object pose estimation and grasp planning. *CoRR*, abs/1809.06893, 2018.

[Bohg *et al.*, 2014] J. Bohg, A. Morales, T. Asfour, and D. Kragic. Data-driven grasp synthesis: A survey. *IEEE Transactions on Robotics*, 30(2):289–309, April 2014.

[Borst *et al.*, 2003] Christoph Borst, Max Fischer, and Gerd Hirzinger. Grasping the dice by dicing the grasp. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 4, pages 3692–3697. IEEE, 2003.

[Bousmalis *et al.*, 2018a] Konstantinos Bousmalis, Alex Irpan, Paul Wohlhart, Yunfei Bai, Matthew Kelcey, Mrinal Kalakrishnan, Laura Downs, Julian Ibarz, Peter Pastor, Kurt Konolige, et al. Using simulation and domain adaptation to improve efficiency of deep robotic grasping. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4243–4250. IEEE, 2018.

[Bousmalis *et al.*, 2018b] Konstantinos Bousmalis, Alex Irpan, Paul Wohlhart, Yunfei Bai, Matthew Kelcey, Mrinal Kalakrishnan, Laura Downs, Julian Ibarz, Peter Pastor, Kurt Konolige, et al. Using simulation and domain adaptation to improve efficiency of deep robotic grasping. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4243–4250. IEEE, 2018.

[Brachmann *et al.*, 2014] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6d object pose estimation using 3d object coordinates. In *European conference on computer vision*, pages 536–551. Springer, 2014.

[Buch *et al.*, 2016] Anders G. Buch, Henrik G. Petersen, and Norbert Krüger. Local shape feature fusion for improved matching, pose estimation and 3d object recognition. *Springerplus*, 5(1):297, 2016.

[Caldera *et al.*, 2018] Shehan Caldera, Alexander Rassau, and Douglas Chai. Review of deep learning methods in robotic grasp detection. *Multimodal Technologies and Interaction*, 2(3):57, 2018.

[Chen and Burdick, 1993] I-Ming Chen and Joel W Burdick. Finding antipodal point grasps on irregularly shaped objects. *IEEE transactions on Robotics and Automation*, 9(4):507–512, 1993.

[Chen *et al.*, 2014] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1971–1978, 2014.

[Chen *et al.*, 2018] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018.

[Chu *et al.*, 2018] Fu-Jen Chu, Ruinian Xu, and Patricio A. Vela. Deep grasp: Detection and localization of grasps with deep neural networks. *CoRR*, abs/1802.00520, 2018.

[Colomé and Torras, 2018] Adrià Colomé and Carme Torras. Dimensionality reduction for dynamic movement primitives and application to bimanual manipulation of clothes. *IEEE Transactions on Robotics*, 34(3):602–615, 2018.

[Crivellaro *et al.*, 2018] Alberto Crivellaro, Mahdi Rad, Yannick Verdie, Kwang M. Yi, Pascal Fua, and Vincent Lepetit. Robust 3d object tracking from monocular images using stable parts. *TPAMI*, 40:1465–1479, 2018.

[Ding *et al.*, 2000] Dan Ding, Yun-Hui Liu, and Shuguo Wang. Computing 3-d optimal form-closure grasps. In *IEEE International Conference on Robotics and Automation*, volume 4, pages 3573–3578. IEEE, 2000.

[Ding *et al.*, 2001] Dan Ding, Yun-Hui Liu, and Michael Yu Wang. On computing immobilizing grasps of 3-d curved objects. In *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 11–16. IEEE, 2001.

[Do *et al.*, 2018] Thanh-Toan Do, Ming Cai, Trung Pham, and Ian Reid. Deep-6dpose: recovering 6d object pose from a single rgb image. *arXiv preprint arXiv:1802.10367*, 2018.

[Dogar *et al.*, 2012] Mehmet Dogar, Kaijen Hsiao, Matei Ciocarlie, and Siddhartha Srinivasa. Physics-based grasp planning through clutter. In *Robotics: Science and Systems VIII*, July 2012.

[Drost and Ilic, 2012] B. Drost and S. Ilic. 3d object detection and localization using multimodal point pair features. In *International Conference on 3D Imaging, Modeling, Processing, Visualization Transmission*, pages 9–16, Oct 2012.

[Drost *et al.*, 2010] B. Drost, M. Ulrich, N. Navab, and S. Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 998–1005, June 2010.

[Ebert *et al.*, 2018] Frederik Ebert, Sudeep Dasari, Alex X. Lee, Sergey Levine, and Chelsea Finn. Robustness via retrying: Closed-loop robotic manipulation with self-supervised learning. *CoRR*, abs/1810.03043, 2018.

[Everingham *et al.*, 2010] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, Jun 2010.

[Everingham *et al.*, 2015] Mark Everingham, S. M. Ali Eslami, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, Jan 2015.

[Fang *et al.*, 2018] Kuan Fang, Yuke Zhu, Animesh Garg, Andrey Kurenkov, Viraj Mehta, Li Fei-Fei, and Silvio Savarese. Learning task-oriented grasping for tool manipulation from simulated self-supervision, 2018.

[Fischer and Hirzinger, 1997] Max Fischer and Gerd Hirzinger. Fast planning of precision grasps for 3d objects. In *IEEE/RSJ International Conference on Intelligent Robot and Systems*, volume 1, pages 120–126. IEEE, 1997.

[Galvez-López and Tardos, 2012] Dorian Galvez-López and Juan D. Tardos. Bags of binary words for fast place recognition in image sequences. *Trans. Rob.*, 28(5):1188–1197, October 2012.

[Garcia-Garcia *et al.*, 2017] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, and Jose Garcia-Rodriguez. A review on deep learning techniques applied to semantic segmentation. *arXiv preprint arXiv:1704.06857*, 2017.

[Girshick *et al.*, 2014] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, pages 580–587, 2014.

[Guo *et al.*, 2016] Di Guo, Tao Kong, Fuchun Sun, and Huaping Liu. Object discovery and grasp detection with a shared convolutional neural network. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2038–2043. IEEE, 2016.

[Hariharan *et al.*, 2011] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *2011 International Conference on Computer Vision*, pages 991–998. IEEE, 2011.

[Haschke *et al.*, 2005] Robert Haschke, Jochen J Steil, Ingo Steuwer, and Helge J Ritter. Task-oriented quality measures for dextrous grasping. In *CIRA*, pages 689–694. Citeseer, 2005.

[He *et al.*, 2017] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask r-cnn. *IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

[Hernandez *et al.*, 2016] Carlos Hernandez, Mukunda Bharatheesha, Wilson Ko, Hans Gaiser, Jethro Tan, Kanter van Deurzen, Maarten de Vries, Bas Van Mil, Jeff van Egmond, Ruben Burger, et al. Team delft's robot winner of the amazon picking challenge 2016. In *Robot World Cup*, pages 613–624. Springer, 2016.

[Hinterstoisser *et al.*, 2012] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *Asian conference on computer vision*, pages 548–562. Springer, 2012.

[Hodaň *et al.*, 2015] Tomáš Hodaň, Xenophon Zabulis, Manolis Lourakis, Štěpán Obdržálek, and Jiří Matas. Detection and fine 3d pose estimation of texture-less objects in rgb-d images. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4421–4428. IEEE, 2015.

[Hodaň *et al.*, 2017] Tomáš Hodaň, Pavel Haluza, Štěpán Obdržálek, Jiří Matas, Manolis Lourakis, and Xenophon Zabulis. T-LESS: An RGB-D dataset for 6D pose estimation of texture-less objects. *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017.

[Hodan *et al.*, 2018a] Tomas Hodan, Rigas Kouskouridas, Tae-Kyun Kim, Federico Tombari, Kostas E. Bekris, Bertram Drost, Thibault Groueix, Krzysztof Walas, Vincent Lepetit, Ales Leonardis, Carsten Steger, Frank Michel, Caner Sahin, Carsten Rother, and Jiri Matas. A summary of the 4th international workshop on recovering 6d object pose. *CoRR*, abs/1810.03758, 2018.

[Hodaň *et al.*, 2018b] Tomáš Hodaň, Frank Michel, Eric Brachmann, Wadim Kehl, Anders GlentBuch, Dirk Kraft, Bertram Drost, Joel Vidal, Stephan Ihrke, Xenophon Zabulis, et al. Bop: benchmark for 6d object pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 19–34, 2018.

[Hu *et al.*, 2018] Yinlin Hu, Joachim Hugonot, Pascal Fua, and Mathieu Salzmann. Segmentation-driven 6d object pose estimation. *arXiv preprint arXiv:1812.02541*, 2018.

[Jiang *et al.*, 2011] Yun Jiang, Stephen Moseson, and Ashutosh Saxena. Efficient grasping from rgbd images: Learning using a new rectangle representation. In *IEEE International Conference on Robotics and Automation*, pages 3304–3311. IEEE, 2011.

[Johnson, 1997] Andrew E Johnson. Spin-images: a representation for 3-d surface matching. 1997.

[Kalashnikov *et al.*, 2018] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *arXiv preprint arXiv:1806.10293*, 2018.

[Kehl *et al.*, 2017] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1521–1529, 2017.

[Kendall *et al.*, 2015] Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. Bayesian segnet: Model uncertainty

in deep convolutional encoder-decoder architectures for scene understanding. *arXiv preprint arXiv:1511.02680*, 2015.

[Krasin *et al.*, 2016] Ivan Krasin, Tom Duerig, Neil Alldrin, Andreas Veit, Sami Abu-El-Haija, Serge Belongie, David Cai, Zheyun Feng, Vittorio Ferrari, and Victor Gomes. Openimages: A public dataset for large-scale multi-label and multi-class image classification., 01 2016.

[Krizhevsky *et al.*, 2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, 2012.

[Kumra and Kanan, 2017] Sulabh Kumra and Christopher Kanan. Robotic grasp detection using deep convolutional neural networks. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 769–776. IEEE, 2017.

[Kwiatkowski and Lipson, 2019] Robert Kwiatkowski and Hod Lipson. Task-agnostic self-modeling machines. *Science Robotics*, 4(26):eaau9354, 2019.

[Laskey *et al.*, 2016] Michael Laskey, Jonathan Lee, Caleb Chuck, David Gealy, Wesley Hsieh, Florian T Pokorny, Anca D Dragan, and Ken Goldberg. Robot grasping in clutter: Using a hierarchy of supervisors for learning from demonstrations. In *IEEE International Conference on Automation Science and Engineering (CASE)*, pages 827–834. IEEE, 2016.

[Lenz *et al.*, 2015] Ian Lenz, Honglak Lee, and Ashutosh Saxena. Deep learning for detecting robotic grasps. *The International Journal of Robotics Research*, 34(4-5):705–724, 2015.

[León *et al.*, 2010] Beatriz León, Stefan Ulbrich, Rosen Diankov, Gustavo Puche, Markus Przybylski, Antonio Morales, Tamim Asfour, Sami Moisio, Jeannette Bohg, James Kuffner, and Rüdiger Dillmann. Opengrasp: A toolkit for robot grasping simulation. In Noriaki Ando, Stephen Balakirsky, Thomas Hemker, Monica Reggiani, and Oskar von Stryk, editors, *Simulation, Modeling, and Programming for Autonomous Robots*, pages 109–120, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[Lepetit *et al.*, 2005] Vincent Lepetit, Pascal Fua, et al. Monocular model-based 3d tracking of rigid objects: A survey. *Foundations and Trends® in Computer Graphics and Vision*, 1(1):1–89, 2005.

[Lepetit *et al.*, 2009] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o(n) solution to the pnp problem. *IJCV*, 81(2):155–166, February 2009.

[Levine *et al.*, 2018] Sergey Levine, Peter Pastor, Alex Krizhevsky, Julian Ibarz, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International Journal of Robotics Research*, 37(4-5):421–436, 2018.

[Li and Sastry, 1988] Zexiang Li and S Shankar Sastry. Task-oriented optimal grasping by multifingered robot hands. *IEEE Journal on Robotics and Automation*, 4(1):32–44, 1988.

[Li *et al.*, 2003] Jia-Wei Li, Hong Liu, and He-Gao Cai. On computing three-finger force-closure grasps of 2-d and 3-d objects. *IEEE Transactions on Robotics and Automation*, 19(1):155–161, 2003.

[Li *et al.*, 2018a] Chunxu Li, Chenguang Yang, Zhaojie Ju, and Andy SK Annamalai. An enhanced teaching interface for a robot using dmp and gmr. *International journal of intelligent robotics and applications*, 2(1):110–121, 2018.

[Li *et al.*, 2018b] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In *Advances in Neural Information Processing Systems*, pages 828–838, 2018.

[Li *et al.*, 2018c] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. Deepim: Deep iterative matching for 6d pose estimation. *Lecture Notes in Computer Science*, pages 695–711, 2018.

[Lin *et al.*, 2014] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing.

[Liu *et al.*, 2004] Yun-Hui Liu, Miu-Ling Lam, and Dan Ding. A complete and efficient algorithm for searching 3-d form-closure grasps in the discrete domain. *IEEE Transactions on Robotics*, 20(5):805–816, 2004.

[Liu *et al.*, 2015] Wei Liu, Andrew Rabinovich, and Alexander C Berg. Parsenet: Looking wider to see better. *arXiv preprint arXiv:1506.04579*, 2015.

[Liu *et al.*, 2016] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[Liu *et al.*, 2018] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. Deep learning for generic object detection: A survey. *arXiv preprint arXiv:1809.02165*, 2018.

[Liu *et al.*, 2019] Fuchang Liu, Pengfei Fang, Zhengwei Yao, Ran Fan, Zhigeng Pan, Weiguo Sheng, and Huansong Yang. Recovering 6d object pose from rgb indoor image based on two-stage detection network with multi-task loss. *Neurocomputing*, 2019.

[Liu, 1999] Yun-Hui Liu. Qualitative test and force optimization of 3-d frictional form-closure grasps using linear programming. *IEEE Transactions on Robotics and Automation*, 15(1):163–173, 1999.

[Long *et al.*, 2015a] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[Long *et al.*, 2015b] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[Lowe, 1999] David G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ICCV '99, pages 1150–, 1999.

[Mahler *et al.*, 2017] Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, and Ken Goldberg. Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. *CoRR*, abs/1703.09312, 2017.

[Miller and Allen, 2004] A. T. Miller and P. K. Allen. Graspit! a versatile simulator for robotic grasping. *IEEE Robotics Automation Magazine*, 11(4):110–122, 2004.

[Miller *et al.*, 2003a] Andrew T. Miller, Steffen Knoop, Henrik I. Christensen, and Peter K. Allen. Automatic grasp planning using shape primitives. In *ICRA*, volume 2, pages 1824–1829, Sep 2003.

[Miller *et al.*, 2003b] Andrew T Miller, Steffen Knoop, Henrik Iskov Christensen, and Peter K Allen. Automatic grasp planning using shape primitives. In *IEEE International Conference on Robotics and Automation*, 2003.

[Mirtich and Canny, 1994] Brian Mirtich and John Canny. Easily computable optimum grasps in 2-d and 3-d. In *IEEE International Conference on Robotics and Automation*, pages 739–747. IEEE, 1994.

[Mottaghi *et al.*, 2014] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, pages 891–898, 2014.

[Mur-Artal *et al.*, 2015] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.

[Nguyen and Le, 2013] Anh Nguyen and Bac Le. 3d point cloud segmentation: A survey. In *2013 6th IEEE conference on robotics, automation and mechatronics (RAM)*, pages 225–230. IEEE, 2013.

[Nguyen, 1987] V-D Nguyen. Constructing stable grasps in 3d. In *IEEE International Conference on Robotics and Automation*, volume 4, pages 234–239. IEEE, 1987.

[Park *et al.*, 2018] Dongwon Park, Yonghyeok Seo, and Se Young Chun. Real-time, highly accurate robotic grasp detection using fully convolutional neural networks with high-resolution images. *arXiv preprint arXiv:1809.05828*, 2018.

[Patil and Rabha, 2018] Aniruddha V Patil and Pankaj Rabha. A survey on joint object detection and pose estimation using monocular vision. *arXiv preprint arXiv:1811.10216*, 2018.

[Perazzi *et al.*, 2016] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 724–732, 2016.

[Pervez and Lee, 2017] Affan Pervez and Dongheui Lee. Learning task-parameterized dynamic movement primitives using mixture of gmms. *Intelligent Service Robotics*, (1):1–18, 2017.

[Pinto and Gupta, 2016] Lerrel Pinto and Abhinav Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3406–3413. IEEE, 2016.

[Ponce *et al.*, 1993] Jean Ponce, Steve Sullivan, J-D Boissonnat, and J-P Merlet. On characterizing and computing three-and four-finger force-closure grasps of polyhedral objects. In *IEEE International Conference on Robotics and Automation*, pages 821–827. IEEE, 1993.

[Pont-Tuset *et al.*, 2017] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017.

[Prats *et al.*, 2007] Mario Prats, Pedro J Sanz, and Angel P Del Pobil. Task-oriented grasping using hand preshapes and task frames. In *IEEE International Conference on Robotics and Automation*, pages 1794–1799. IEEE, 2007.

[Qi *et al.*, 2017] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017.

[Quillen *et al.*, 2018] Deirdre Quillen, Eric Jang, Ofir Nachum, Chelsea Finn, Julian Ibarz, and Sergey Levine. Deep reinforcement learning for vision-based robotic grasping: A simulated comparative evaluation of off-policy methods. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6284–6291. IEEE, 2018.

[Rad and Lepetit, 2017] Mahdi Rad and Vincent Lepetit. Bb8: a scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. In *IEEE International Conference on Computer Vision*, pages 3828–3836, 2017.

[Rai *et al.*, 2017] Akshara Rai, Giovanni Sutanto, Stefan Schaal, and Franziska Meier. Learning feedback terms for reactive planning and control. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2184–2191. IEEE, 2017.

[Redmon and Angelova, 2015] Joseph Redmon and Anelia Angelova. Real-time grasp detection using convolutional neural networks. In *2015 IEEE International Conference*

*on Robotics and Automation (ICRA)*, pages 1316–1322. IEEE, 2015.

[Redmon *et al.*, 2016] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[Ren and Sudderth, 2018] Zhile Ren and Erik B Sudderth. 3d object detection with latent support surfaces. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 937–946, 2018.

[Rennie *et al.*, 2015] Colin Rennie, Rahul Shome, Kostas E. Bekris, and Alberto F. De Souza. A dataset for improved rgbd-based object detection and pose estimation for warehouse pick-and-place. *CoRR*, abs/1509.01277, 2015.

[Roa and Suárez, 2015] Máximo A Roa and Raúl Suárez. Grasp quality measures: review and performance. *Autonomous robots*, 38(1):65–88, 2015.

[Russakovsky *et al.*, 2015] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, Dec 2015.

[Rusu *et al.*, 2009] R. B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (fpfh) for 3d registration. In *IEEE International Conference on Robotics and Automation*, pages 3212–3217, May 2009.

[Sahbani *et al.*, 2012] A. Sahbani, S. El-Khoury, and P. Bidaud. An overview of 3d object grasp synthesis algorithms. *Robotics and Autonomous Systems*, 60(3):326 – 336, 2012. Autonomous Grasping.

[Salti *et al.*, 2014] Samuele Salti, Federico Tombari, and Luigi Di Stefano. Shot: Unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding*, 125:251 – 264, 2014.

[Sanchez *et al.*, 2018] Jose Sanchez, Juan-Antonio Corrales, Belhassen-Chedli Bouzgarrou, and Youcef Mezouar. Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey. *The International Journal of Robotics Research*, 37(7):688–716, 2018.

[Schaal, 2006] Stefan Schaal. Dynamic movement primitives-a framework for motor control in humans and humanoid robotics. In *Adaptive motion of animals and machines*, pages 261–280. Springer, 2006.

[Simon *et al.*, 2018] Martin Simon, Stefan Milz, Karl Amende, and Horst-Michael Gross. Complex-yolo: An euler-region-proposal for real-time 3d object detection on point clouds. In *European Conference on Computer Vision*, pages 197–209. Springer, 2018.

[Song and Xiao, 2014] Shuran Song and Jianxiong Xiao. Sliding shapes for 3d object detection in depth images. In *European conference on computer vision*, pages 634–651. Springer, 2014.

[Song *et al.*, 2018] Peng Song, Zhongqi Fu, and Ligang Liu. Grasp planning via hand-object geometric fitting. *The Visual Computer*, 34(2):257–270, 2018.

[Sundermeyer *et al.*, 2018] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3d orientation learning for 6d object detection from rgb images. In *European Conference on Computer Vision.*, pages 712–729. Springer International Publishing, 2018.

[Tejani *et al.*, 2014] Alykhan Tejani, Danhang Tang, Rigas Kouskouridas, and Tae-Kyun Kim. Latent-class hough forests for 3d object detection and pose estimation. In *European Conference on Computer Vision*, pages 462–477. Springer, 2014.

[Tekin *et al.*, 2018] Bugra Tekin, Sudipta N Sinha, and Pascal Fua. Real-time seamless single shot 6d object pose prediction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 292–301, 2018.

[ten Pas *et al.*, 2017] Andreas ten Pas, Marcus Gualtieri, Kate Saenko, and Robert Platt. Grasp pose detection in point clouds. *Int. J. Rob. Res.*, 36(13-14):1455–1473, December 2017.

[Tian *et al.*, 2018] Hao Tian, Changbo Wang, Dinesh Manocha, and Xinyu Zhang. Transferring grasp configurations using active learning and local replanning, 2018.

[Vacchetti *et al.*, 2004] Luca Vacchetti, Vincent Lepetit, and Pascal Fua. Stable real-time 3d tracking using online and offline information. *IEEE transactions on pattern analysis and machine intelligence*, 26(10):1385–1391, 2004.

[Vahrenkamp *et al.*, 2016] Nikolaus Vahrenkamp, Leonard Westkamp, Natsuki Yamanobe, Eren E Aksoy, and Tamim Asfour. Part-based grasp planning for familiar objects. In *IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 919–925. IEEE, 2016.

[Vidal *et al.*, 2018] J. Vidal, C. Lin, and R. Martí. 6d pose estimation using an improved method based on point pair features. In *2018 4th International Conference on Control, Automation and Robotics (ICCAR)*, pages 405–409, April 2018.

[Viereck *et al.*, 2017] Ulrich Viereck, Andreas ten Pas, Kate Saenko, and Robert Platt. Learning a visuomotor controller for real world robotic grasping using simulated depth images. *arXiv preprint arXiv:1706.04652*, 2017.

[Wang *et al.*, 2019a] Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martín-Martín, Cewu Lu, Li Fei-Fei, and Silvio Savarese. Densefusion: 6d object pose estimation by iterative dense fusion. *arXiv preprint arXiv:1901.04780*, 2019.

[Wang *et al.*, 2019b] Tao Wang, Chao Yang, Frank Kirchner, Peng Du, Fuchun Sun, and Bin Fang. Multimodal grasp data set: A novel visual–tactile data set for robotic manipulation. *International Journal of Advanced Robotic Systems*, 16(1):1729881418821571, 2019.

[Wong *et al.*, 2017] Jay M Wong, Vincent Kee, Tiffany Le, Syler Wagner, Gian-Luca Mariottini, Abraham Schneider, Lei Hamilton, Rahul Chipalkatty, Mitchell Hebert, David MS Johnson, et al. Segicp: Integrated deep semantic segmentation and pose estimation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5784–5789. IEEE, 2017.

[Xiang *et al.*, 2017] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017.

[Xiao *et al.*, 2016] Jianxiong Xiao, Krista A. Ehinger, James Hays, Antonio Torralba, and Aude Oliva. Sun database: Exploring a large collection of scene categories. *International Journal of Computer Vision*, 119(1):3–22, Aug 2016.

[Xu *et al.*, 2018] Danfei Xu, Dragomir Anguelov, and Ashesh Jain. Pointfusion: Deep sensor fusion for 3d bounding box estimation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018.

[Xue *et al.*, 2009] Zhixing Xue, Alexander Kasper, J Marius Zoellner, and Ruediger Dillmann. An automatic grasp planning system for service robots. In *2009 International Conference on Advanced Robotics*, pages 1–6. IEEE, 2009.

[Yang *et al.*, 2018] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7652–7660, 2018.

[Yu and Koltun, 2015] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[Zeng *et al.*, 2017] Andy Zeng, Kuan-Ting Yu, Shuran Song, Daniel Suo, Ed Walker, Alberto Rodriguez, and Jianxiong Xiao. Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1386–1383. IEEE, 2017.

[Zeng *et al.*, 2018a] Andy Zeng, Shuran Song, Stefan Welker, Johnny Lee, Alberto Rodriguez, and Thomas Funkhouser. Learning synergies between pushing and grasping with self-supervised deep reinforcement learning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[Zeng *et al.*, 2018b] Andy Zeng, Shuran Song, Kuan-Ting Yu, Elliott Donlon, Francois R Hogan, Maria Bauza, Daolin Ma, Orion Taylor, Melody Liu, Eudald Romo, et al. Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8. IEEE, 2018.

[Zheng *et al.*, 2015] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1529–1537, 2015.

[Zhou *et al.*, 2018] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, June 2018.

[Zhu and Wang, 2003] Xiangyang Zhu and Jun Wang. Synthesis of force-closure grasps on 3-d objects based on the q distance. *IEEE Transactions on robotics and Automation*, 19(4):669–679, 2003.