Robotic Autonomous Grasping Technique: A Survey

Lili Wang
[1]the School of Artificial Intelligence,
University of Chinese Academy of Sciences
[2]the State Key Lab. of Management and
Control for Complex Systems,
Institute of Automation, Chinese Academy of Sciences
Beijing, P.R. China.
wanglili2020@ia.ac.cn

Zhen Zhang
[1]the School of Artificial Intelligence,
University of Chinese Academy of Sciences
[2]the State Key Lab. of Management and
Control for Complex Systems,
Institute of Automation, Chinese Academy of Sciences
Beijing, P.R. China.
zhangzhen2020@ia.ac.cn

Jianhua Su*
[1]the School of Artificial Intelligence,
University of Chinese Academy of Sciences
[2]the State Key Lab. of Management and
Control for Complex Systems,
Institute of Automation, Chinese Academy of Sciences
Beijing, P.R. China.
jianhua.su@ia.ac.cn

Qipeng Gu
[1]the School of Artificial Intelligence,
University of Chinese Academy of Sciences
[2]the State Key Lab. of Management and
Control for Complex Systems,
Institute of Automation, Chinese Academy of Sciences
Beijing, P.R. China.
guqipeng2019@ia.ac.cn

2021 5th Asian Conference on Artificial Intelligence Technology (ACAIT) | 978-1-6654-2630-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ACAIT53529.2021.9731320

***Abstract*—This paper provides a comprehensive survey of robotic autonomous grasping techniques. We summarize three key tasks: grasp detection, affordance detection, and model migration. Grasp detection determines the graspable area and grasping posture of the manipulator, so that the robot can successfully perform the grasps. The grasp detection methods based on deep learning are divided into 3DoF grasp and 6DoF grasp. The object affordances based grasping methods can further improve the robot's understanding of objects and environment, thereby improving the robot's intelligence and autonomy. Methods for object affordances detection are classified as learning-based, knowledge-based, and simulation-based. Model migration means that when the grasping model is migrated to other scenes where lightness and background changes, only little or no label data is required, so that the grasping model can be used in the target scene quickly and efficiently. This paper focuses on domain adaptation (DA) methods in model migration.**

***Index Terms*—Robotic grasping, Affordance detection, Domain adaptation**

I. Introduction

The stable and reliable grasp is fundamental and significant for robots to complete assembly, handling and sorting tasks. Robotic grasp detection is a key research component in the field of robotic autonomous grasping, which determines the graspable area on the object and grasp pose of the robot end-effector. Early grasp detection focuses on object geometry, physical model, kinematics and mechanical analysis [1]. Such as searching for contact points that afford the form and force closure on objects of known three-dimensional models. However, these approaches are often computationally unaffordable and not adapted for new tasks or novel objects. Most recently, the use of deep neural network methods to train an end-to-end grasping strategy has made great progress. High-quality grasps generated by deep learning have anti-interference ability and strong generalization ability. This paper focuses on the grasp detection based on deep learning.

In spite of the development of robotic grasp detection, the grasp of arbitrary objects in unstructured environments is still a challenging and complex task. Robot vision aims to discover and understand information, then interact with the environment. This requires the robot to understand the affordances of objects even in the complex visual domains. Affordances refer to the properties or characteristics of objects that provide the agent with a series of potential actions. In other words, this research field explores how robots use objects. Gibson first proposed the concept of "affordances" in 1966 [2]. Since then, affordance detection has been widely applied to perform higher-level reasoning on the scene. We review the affordance detection approaches in this paper.

Although the robotic grasping technology based on deep learning has been extensively studied, it still has many limitations. Deep learning demands a great deal of learning data, and collecting datasets and labelling are time-consuming and labor-intensive. So most methods are trained on public datasets, this leads to poor results when the model is migrated to other scenes with different grasping background, view angle, lightness, sensor, etc. Robotic grasping based on deep learning is quite domain-related, studying the migration of the model trained in the source domain to the target domain with less annotated data is necessary to bridge the gap between artificial intelligence and practical applications.

The rest of this paper is arranged as follows. Section II

This work was supported in part by NSFC under Grant number 91848109, supported by Beijing Natural Science Foundation under Grant number L201019 and supported by Science and Technology on Space Intelligent Control Laboratory under HTKJ2019KL502013.

978-1-6654-2630-5/21/$31.00 ©2021 IEEE

reviews the deep learning based grasping detection algorithms. Section III reviews the methods for object affordances in robotic grasping. Section IV reviews the methods for grasping detection model migration. Finally, the conclusion is drawn in Section V.

II. Methods for Grasping Detection Based on Deep Learning

To grasp object, the 6-dimensional pose of robot end-effector in the camera coordinate is necessary information. In this paper, the robot end-effector we only talk about parallel grippers. The grasping configuration describes how to arrange the gripper 6D pose to successfully grasp the object. According to the different grasping configuration, the grasping detection methods based on deep learning can be categorized into the 3DoF grasp and 6DoF grasp.

A. 3DoF Grasp

3DoF grasp refers to the grasp pose that contains a two-dimensional in-plane location and a one-dimensional rotation angle. 3DoF grasp is also called 2D plane grasp because the grasp pose is limited by the direction which is perpendicular to the workspace plane. 2D planer grasp methods can be divided into structured grasping configuration and pixel-level grasping configuration.

1) Structured Grasping Configuration: In the early research, the grasping configuration is based on points on scene images. Aiming to find graspable points in the discrete three-dimensional space, Saxena et al. [3] proposed a regression learning method to estimate the graspable point position in the cartesian coordinate system. But this approach only determines where to grasp, not determines the gripper orientation. To overcome this limitation, the oriented rectangle is proposed to represent the grasping configuration and has been extensively studied. Jiang et al. [4] proposed the use of directed rectangle containing 3D position, 3D orientation and the gripper opening width, expressing as $G = (x, y, z, \alpha, \beta, \gamma, w)$ to estimate 7-dimensional grasp. Such presentation brings computational expensively. Lenz et al. [5] used the rectangle with location, orientation and size: $G = (x, y, \theta, h, w)$ to simplify the above-mentioned grasping configuration from 7D to 5D. Fig. 1 shows an example of rectangle presentation. This paper is the first to use deep learning in the field of robotic grasping. They proposed to use two networks as a two-step cascaded system, first effectively pruning out impossible candidate grasps and then re-evaluating the rest of grasps to find top-ranked rectangle with a larger network. This 5D rectangle grasp representation is used in many subsequent studies. Redmon et al. [6] addressed the same problem as Lenz [5] with the 5D configuration, but used a different network architecture that performs single-stage regression from RGB-D image to graspable bounding boxes, and achieves the faster and more accurate performance. Chu et al. [7] applied 5D configuration to the multiple object situations. They proposed a framework for predicting multiple grasp candidates rather than a single outcome in a single shot. This framework transforms orientation regression to a classification problem and predicts both grasp regression values and discrete orientation classification scores. Pinto et al. [8] used the oriented rectangle only with the position and orientation: $G = (x, y, \theta)$, removed the rectangle size parameters. They recast the regression to a binary classification and used a CNN-based classifier to predict the grasp possibility for different grasp directions.

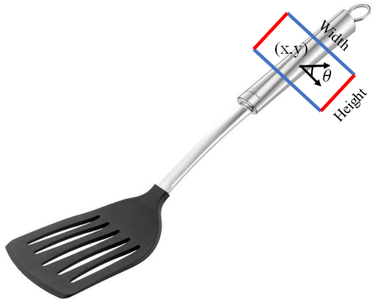

Fig. 1. A five-dimensional rectangular grasp presentation with position, size, and orientation.

Another structured grasping configuration is grasping contact points which uniquely defines the grasping pose. Mahler et al. [9] used the point with position and angle in the table plane, expressing as: $G = (x, y, z, \theta)$. They proposed a network architecture to predict the robustness scores of candidate grasps, and then selected the highest quality one as the final grasp. Learning the grasp robustness function is one of the central ideas of deep learning based grasp detection research. It describes the successful grasp possibility of a location or region in the image, and identifies the highest-scoring grasp candidate as the output.

2) Pixel-level Grasping Configuration: The pixel-level grasping configuration is to estimate the grasp quality for each pixel in the image or to estimate pixel-wise grasp affordances to evaluate the most probable grasping contact points. Morrison et al. [10] proposed Generative Grasping Convolutional Neural Network achieving one-to-one mapping from a depth image to grasp map, which consists of three pixel images: grasp quality, grasp angle, and grasp width. These three pixel images determine a grasp at each pixel. This network is light-weight and fast. Zeng et al. [11] presented a framework achieving pixel-wise grasp affordances predictions that returns grasp locations, orientations, and the confidence scores. The grasp affordances are pre-defined as parallel gripper grasp and suction cup grasp. The author used two fully convolutional networks to predict this two affordances under 16 different angles to judge whether the object can be sucked or has a graspable area, and then generated grasp proposals with confidence scores. Cai et al. [12] used network as an affordance interpreter to predict pixel-wise grasp affordance map, where each pixel belongs to one of

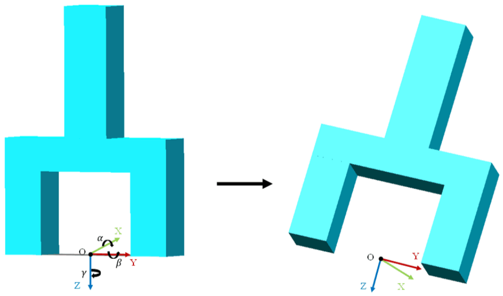

Fig. 2. The left shows the coordinate frame of the gripper. The right shows the gripper rotates around the three coordinate axes.

predefined grasp affordances. These two papers predict pixel-wise affordance maps and directly estimate grasp qualities. [11], [12] use grasp affordance detection that does not involve object understanding, that is, they only use the predefined grasp affordances which aim to find the most suitable grasping area. In section III, the methods of object affordances detection will be introduced.

B. 6DoF Grasp

6DoF grasp means the gripper's 6D pose in camera coordinate should be all estimated to allow grasping objects from various angles, which is shown in Fig. 2. 6D pose includes 3D position and 3D orientation of the gripper. The early analytical methods analyzed the geometric structure of known object 3D model to determine the graspable points that satisfy some certain quality metrics [1]. As depth image becomes easily available, monocular object 6D pose estimation [13] [14] is extensively researched. With object complete shape, the object 6D pose can be achieved and 6DoF grasp can be inferred. If the target object is known and 6DoF grasp poses are precomputed, the 6DoF grasps can be obtained through sampling and ranking the grasp poses in the knowledge base, and then the problem of estimating grasp poses is converted into the problem of estimating object 6D pose. Deng et al. [15] predict the objects 6D poses based on the prior knowledge about objects shape and then project the predefined grasp poses to the workspace.

From [16], there is a new research direction that based on partial point cloud which requires no prior knowledge about objects, merely analysis the input partial point cloud to estimate the 6DoF grasp poses. Most of these methods propose grasp candidates and estimate the grasp quality for each candidate. ten Pas et al. [16] proposed GPD algorithm that first samples 6DoF grasp candidates from a region of interest (ROI) which is identified by preprocessed viewpoint cloud, then the candidates are encoded as a stacked multi-channel image. Use convolutional neural network (CNN) to evaluated the each candidate score, then select a grasp for execution based on this score. Liang et al. [17] made further expansion and proposed PointNetGPD. Instead of multi-view projection features, take raw point cloud as input. Then evaluate the quality of the sampled candidate grasps through geometric analysis based on the PointNet [18]. This work outperforms GPD when input point cloud is sparse overall. Mousavian et al. [19] proposed 6DoF GraspNet algorithm that uses a variational autoencoder to sample grasps and then uses a grasp evaluator model to assess and refine the sampled grasps. Qin et al. [20] proposed a single-shot grasp network based on PointNet++ [21]. This is a direct regression method for predicting 6DoF grasps, and each grasp has a grasp quality score to evaluate.

III. Methods for Object Affordances Detection in Robotic Grasping

Affordance specifies the functions that the object allows to the user (or agent), that is, what operations the user can perform on a given object in the environment. Ecological psychologist Gibson first introduced the concept of affordances in 1966 [2]. Humans use vision to easily acquire object affordances and utilize this information to perform daily tasks including grasping objects. From the priori experience, humans can determine the best way to grasp. In robotics, detecting objects affordances is a fundamental ability for robots to understand the objects. The grasp detection methods mentioned in the previous section can successfully execute the grasping action, but can not make the robots perform tasks like human. So efficient affordance detection is the core function in the developing autonomous systems. Vision-based affordance is a branch of the field of computer vision and a detailed review about visual affordance can be found in [22]. In the following, we survey the affordance detection based grasping approaches.

A. Learning-based Affordance Detection

Since humans mainly use visual cues to reason about the object affordances, there are many studies using RGB-D images modeling. Collect data by using human knowledge and learn object affordances from it. Due to the popularity of deep learning, many researches use CNN to replace traditional feature engineering. And affordance detection is regarded as the problem of labelling parts of objects at pixel level by function. Nguyen et al. [23] proposed a deep CNN-based encoder-decoder architecture to detect object affordances. This method used automatic feature learning instead of manual features. The depth image is encoded into three channels and these channels combine with RGB images to form multiple modalities input data. The output is a probability image, the number of channels is the number of affordance classes. They tested this algorithm on a real robot conducting grasping and got significant enhancements on the UMD dataset [24]. But the object scene in this dataset is not occluded or cluttered. Another work also by Nguyen et al. [25] treats the affordance detection as an object detection problem. They used a deep network to predict the location of objects and represent it with bounding box. Then deep CNN is used to create feature maps from these bounding boxes, and finally these feature maps are processed using Conditional Random Fields model to further enhance the prediction results of affordance label of each pixel. Based on the detected affordances, they conducted real robot to perform grasp and demonstrated higher success rate and

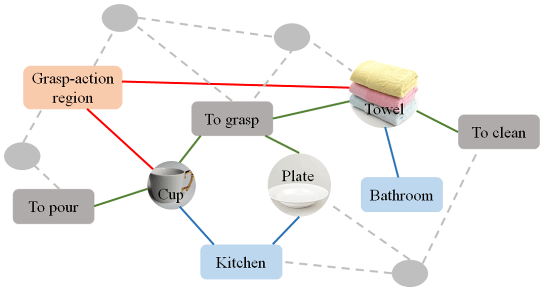

Fig. 3. An example knowledge graph. Relevant nodes are interconnected, different color edges depict different relations between nodes, such as attributes, affordances, and graspable area.

more robustness. Do et al. [26] proposed a AffordanceNet which achieves the simultaneous detection of multiple objects and their affordances. The AffordanceNet has two branches, one for object detection and one for affordance detection. Significantly, the end-to-end model learning can perform model training in a single framework, which has recently dominated recognition techniques. Chu et al. [27] proposed a framework that simultaneously detects objects position in the image, object labels, and multi-label affordance maps. But unlike the supervised learning described above, this framework takes into account the adaptation of the domain from synthetic data to real images and avoids the need for large annotated data when model is migrated to other domains. We will introduce domain adaptation in detail in section IV. In addition to studies in 2.5D image domains, Deng et al. [28] presented a 3D AffordanceNet dataset and focused on visual affordance detection on point cloud data.

B. Knowledge-based Affordance Detection

The knowledge base (KB) is a library composed of entities and rules, which is used to store and query the affordance of objects. KB can be considered as a graph, where the notes represent entities and edges represent general rules. Every entity in the KB consists of object attributes and affordances. Fig. 3 shows a KB example. A learned KB is a unified framework under which many different reasoning tasks can be performed without any further training. Zhu et al. [29] extracted object information including their attributes and affordances from images and online textual sources. Each object has three attributes: visual attributes, physical attributes, and categorical attributes. These attributes allow knowledge to be transferred between objects, thus enabling the prediction of the novel object affordance. After data collection, use Markov Logic Network (MLN) to learn relations from it to construct a knowledge graph. When performing reasoning, the model first extracts the object visual attributes based on the image, and then infers physical and categorical attributes, finally queries the object affordances in the acquired knowledge graph. Ardón et al. [30] proposed a method for detecting and extracting multiple grasp affordances on one object. This work is similar to that of [29], but focuses on solving the inference of the grasp affordances which is subdivided. They used MLN to obtain semantics relationships between attributes, locations and grasp affordances to build a KB. In the affordance reasoning stage, a CNN is used to extract the objects' attributes from RGB images, and use Gibbs sampling [31] to query the approximation of the probability distribution related to grasp affordances from the learned model, so as to obtain the most probable grasp affordance. The method of building a knowledge graph is to use multiple clues to complete affordance detection. The model is robust, interpretable, and easy to extend. But the quality of the model largely depends on the knowledge graph, which in turn depends on the quality of the collected data.

C. Simulation-based Affordance Detection

In contrast to learning appearance-based cues and building knowledge base, there are some simulation-based approaches to encode object affordances. In [32], using a physical simulations, particles are dropped onto the object and the number of particles left in it to quantify the open containability affordance. First, a robot with a in-hand RGB-D camera acquires object 3D model by scanning it. Then use the 3D model for open containability and pouring imagination. On the open container classification, this method's performances is comparable to deep learning approaches, but outperforms it on autonomous pouring. Abelha et al. [33] made a robot find the best way to grasp and orient an object. They gathered 3D models from the internet and through simulation automatically learned both a object's affordances and how the object should be held and oriented. These methods dig into the underlying physics of the object to obtain the affordanes and avoid the problem that the appearance-based method become fragile when meating objects with large intra-class variations. Simulation-based method can realize functional reasoning of objects that have not been seen before well and achieve the inter-class function generalization.

IV. Methods for Grasping Detection Model Migration

Robot autonomous grasping based on deep learning methods is a hot research content. Training grasp detection and affordance detection models based on deep learning are both supervised learning which is in demand of large number of well-labeled data, and models are trained under the assumption that both the training and testing set share the same distribution. However, collecting data and making them annotated are quite time-consuming. In some extreme working environment, it is quite hard to acquire a well-labeled dataset. Besides, testing set may have feature space different from that of training set or follow a different distribution. In this scenario, we need to deal with the problem of lacking sufficient well-labeled data.

A common scenario in grasping detecting task is that there are plenty well-labeled instances on public dataset which is easy to obtain. While in new tasks, we may manage to acquire only limited number of labeled instances that are different from instances on the public dataset, or worse, there are only instances without label. Luckily, the task remain the same, that

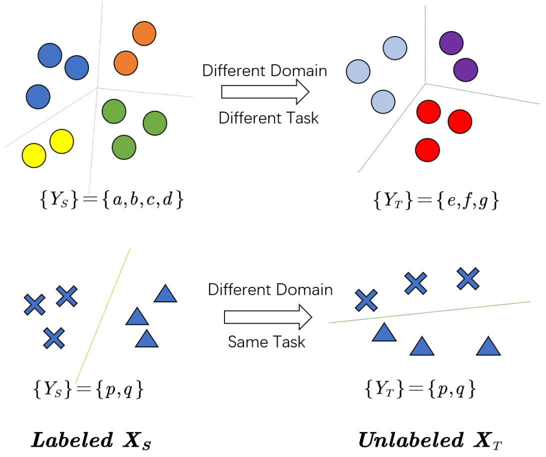

Fig. 4. The difference between transfer learning (upper) and domain adaptation (lower). Domains and tasks may both be different in transfer learning, while in domain adaptation, when the distribution of X changes, the task remain the same.

is, the public dataset and task at hand share the same label set, the marginal distribution or conditional distribution is the only place where they differ. Thus, domain adaptation is brought up to handle this situation.

A. A Brief Introduction to Domain Adaptation

As a matter of fact, it usually considered that DA is one special branch of transfer learning [34], as shown in Fig. 4. In transfer learning, a domain D, defined as $D = \{\mathcal{X}, P(X)\}$, consists of a feature space $\mathcal{X}$ and a marginal distribution $P(X)$. Here X refers to an instance set in which each sample is denoted as $\mathbf{x}_i, i = 1, \ldots, n$. And a task $\mathcal{T}$, defined as $\mathcal{T} = \{\mathcal{Y}, f\}$, consists of a decision function f which can map each x_i from X into its corresponding label y_i in the label space $\mathcal{Y}$.

Given observations $D_S = (X_S, Y_S)$ and $D_T = (X_T)$ where Y_T, the corresponding labels of X_T is unknown, domain adaptation assume that two domains share the same task, i.e., $\{Y_T\} = \{Y_S\}$ or $\mathcal{Y}_T = \mathcal{Y}_S$, but instances drawn from different domain follow different marginal distribution, i.e., $P(X_T) \neq P(X_S)$. So DA methods tries to leverage knowledge learned from D_S to acquire a decision function f_T to perform well in D_T obtain $\hat{Y}_T$.

According to [34], transfer learning problems can be classified as transductive, inductive and unsupervised transfer learning. In the first category, the target domain shares the same label space with source domain, i.e., $\mathcal{Y}_T = \mathcal{Y}_S$, but has no label available. Since there are abundant well-annotated data available in D_S, we can learn a basic decision function f_S and leverage the knowledge implied in X_T to make f_S perform better on target domain. When target domain has different task, this situation is categorized into inductive transfer learning. In this situation, the target domain may have a different label space compared with that of source domain, hence in target domain, labeled data is needed so as to make the task learnable. Further on, when labels on both domains are no longer available, then it becomes an unsupervised transfer learning problem. Different with inductive transfer learning, it mainly tackles the unsupervised learning problem on the target domain, such as clustering and estimation of probability density .

Different domain of transfer learning may have different task $\mathcal{T}$, and a similar thought of jointly learning multiple tasks is implemented in multitask learning. Multitask learning learns a group of related tasks which share intertask information. Since it hypothesize that related tasks are able to use intertask information, multitask learning is able to learn and at the same time keep the inner structure of data, and only transfer shared knowledge expression among all the tasks. Transfer learning and multitask learning adopt some similar methods so as to leverage the transferred knowledge. But they also hold some differences, multitask learning learns a group of tasks simultaneously, while tasks of target domain will acquire more attention in transfer learning.

Multiview learning aims to learn multiview data, that is, to learn data with multiple set of features, such as a video object with both image features and audio features. Multiview learning holds the belief that multiple views of the same data contain complementary information, and by describing information from multiple views of the given data, the learner can master a more comprehensive and more compact expression of the data, and hence achieve better performance. Sophisticated application includes recommender system [35], video analysis [36], and natural language processing [37].

Domain generalization [38], [39] tries to train model on several labeled domains. After training, it will devote to generalize them to unseen domains. Different with DA methods, in the training process, data in target domain are not available, while domain adaptation still needs them to adjust for cross-domain migration.

Of course, there are also domain adaptation methods used in reinforcement learning to generate robust grasping strategies, but this section mainly discusses the methods that can be used in domain adaptation methods to generate grasping configurations in robot grasp detection from a visual perspective.

B. Categories of Domain Adaptation

The existing DA methods can be categorized as shallow architectures and deep architectures. Shallow domain adaptation mainly aims to align domain distribution. One way to achieve this goal is to minimize the distance between different domains. Most commonly seen metrics include the maximum mean difference (MMD) [40], the Correlation Alignment [41], Kullback-Leibler(KL) divergence [42] and Contrastive Domain Discrepencies (discrepencies in pairs, CDD) [43]. Deep domain adaptation algorithm uses deep neural network. Such methods usually use convolution, autoencoders, or GAN [44] to reduce the distance between domains.

1) Shallow Domain Adaptation: Traditional machine learning methods minimize the loss defined as follow:

$$\min_{f} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(f(\mathbf{x}_i), \mathbf{y}_i) \tag{1}$$

where $\mathbf{x}_i$ denotes the input features of the i-th instance, $\mathbf{y}_i$ demotes the corresponding label, f is our desired decision function which maps the input feature to the label space, $\mathcal{L}$ is the loss to be minimized and n is the total number of instances.

In unsupervised domain adaptation(UDA), target domain has no labeled data available. So we need to find approaches to transfer the knowledge implied in the source domain.

a) Instance-Based Approaches: Considering that there are always some instances that are very similar in the source domain and the target domain, then the loss of all instances from the source domain is multiplied by a weight during training.

$$\min_{f} \frac{1}{n^S} \sum_{i=1}^{n^S} w_i \mathcal{L}(f(\mathbf{x}_i^S), \mathbf{y}_i^S) \tag{2}$$

where $\mathbf{x}_i^S$ refers to the input features of the i-th instance, $\mathbf{y}_i^S$ demotes the corresponding label. w_i describes the similarity between the i-th instance and instances from target domain, that is, the more similar the i-th instance is to the target domain, the greater the weight w_i will be, n^S is the total number of instances from the source domain.

It is easy to come up with the idea that we can adapt the marginal distributions to obtain w, and the weighting strategy is thus follows equation [45]:

$$\begin{aligned}\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim P^T}[\mathcal{L}(f(\mathbf{x}), \mathbf{y})] &= \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim P^S}[\frac{P^S(\mathbf{x},\mathbf{y})}{P^S(\mathbf{x},\mathbf{y})}\ \mathcal{L}(f(\mathbf{x}), \mathbf{y})] \\ &= \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim P^S}[\frac{P^T(\mathbf{x})}{P^S(\mathbf{x})}\ \mathcal{L}(f(\mathbf{x}), \mathbf{y})]\end{aligned} \tag{3}$$

where P^S,P^T are distribution of the each domain. Hence the theoretical value of w_i should be $\frac{P^T(\mathbf{x}_i)}{P^S(\mathbf{x}_i)}$. There are many methods to estimate this theoretical ratio. Kernel mean matching (KMM) [45] resolves the estimation problem by minimizing the MMD metric in a reproducing kernel Hilbert space (RKHS) between instances drawn from different domains. KL importance estimation procedure (KLIEP) [46] is another way which mainly minimizes the KL divergence to obtain the estimated ratio.

b) Feature-Based Approaches: Feature-based approaches assume that features of instances in both domains can be mapped into a common feature space in which features from different domain can be aligned. That is to say, these approaches try to learn a transformation that represents the original features as domain-invariant features, and at the same time keep the inner structure of the original data. The general learning object is as follow:

$$\min_{f} \frac{1}{n^S} \sum_{i=1}^{n^S} \mathcal{L}(f(\phi(\mathbf{x}_i^S)), \mathbf{y}_i^S) \tag{4}$$

where ϕ is the mapping function.

Transfer Component Adaptation (TCA) [47] projects features of both domains into a RKHS, and minimizes the marginal distribution of different domain by minimizing the MMD metric of points of two domains in the RKHS to find desired domain-invariant features. Geodesic Flow Kernel (GFK) [48] and Sampling Geodesic Flow (SGF) [49] first perform PCA to obtain subspaces of two domains, and then view two obtained subspaces as two points on the Grassmann manifold [50], they then utilize the potential path between the two points, and obtain doamin-invariant features by stacking the projections from all the subspaces generated by interpolating between two points on the manifold based on properties of the Grassmann manifold. Jhuo et al. [51] show us that a linear projection matrix which can transform source-domain instances into a meta representation in which source-domain instance can be linearly represented by target-domain instances, so as to align the features, can be learned to perform well as a novel DA method.

2) Deep Domain Adaptation: Since deep networks always outperform traditional approaches based on hand-crafted features on most discriminant tasks, it is natural to think that introducing deep network into DA can also greatly enhance the performance.

Deep DA approaches have three types: the discrepancy-based, the adversarial-based and the reconstruction-based.

a) Discrepancy-Based Approaches: Deep domain confusion (DDC) [52] adds an adaptation deep neural network together with a novel discrepancy loss, while extracting features from both domains, it minimizes the classification loss and discrepancy loss at the same time.

$$\mathcal{L} = \mathcal{L}_C(f(\mathbf{x}^S), \mathbf{y}^S) + \lambda\ \mathcal{L}_D(\phi(\mathbf{x}^S), \phi(\mathbf{x}^T)) \tag{5}$$

ϕ here is introduced as a representation of feature extraction function, and f denotes the predict function of the whole network, $\mathcal{L}_C$ denotes the classification loss and $\mathcal{L}_D$ denotes the discrepancy loss of the input features of the two domains, which is acquired by calculating the MMD of $\phi(\mathbf{x}^S)$ and $\phi(\mathbf{x}^T)$.

Deep Adaptation Network (DAN) [53] adds multiple adaptation layers and exploring multiple kernels, each adaptation layer has its own discrepancy loss. Different with DDC, here the discrepancy loss is multi-kernel version of MMD (MK-MMD) [54] which takes advantages of several kernels.

b) Adversarial-Based Approaches: GAN [44] introduced the game theory into deep learning. Generally, it plays a minimax game to let a generator be able to generate features that can confuse the discriminator. Motivated by GAN, it natural to draw the conclusion that we can obtain domain-invariant features representation by playing the similar minimax game.

Domain-adversarial neural network (DANN) [55] uses a feature extractor which acts like generator in GAN to extract deep domain-invariant features from both domains, and uses a domain classifier as the discriminator to detect which domain the extracted features come from. A special layer termed gradient reversal layer (GRL) is plugged into the base model to train faster. After training, a simple downstream label predictor can well perform in the target domain using its extracted domain-invariant features.

Tzeng et al. [56] proposed a model named ADDA which unties the weights of feature extractors of two domain, that is, each has its own feature extractor. This makes ADDA able to leverage more domain-specific features. To be mentioned with, in their earlier work [57], they showed a method which adds soft label loss to align not only marginal but also conditional distribution of both domains.

c) Reconstruction-Based Approaches: Another way of align the features of different domains is to learn to reconstruct the instance and minimize the reconstruction error. And the reconstruction can be achieved by autoencoders [58] or GAN. Stacked denoising autoencoders (SDA) [59] uses autoencoders for reconstruction. The denoising autoencoder is an extension of the original autoencoder which will add noise to the input. This corrupting mechanism is proved to enhance the robustness. In SDA, instances from both domains are used to train multiple denoising autoencoders, hence desired domain-invariant features representation can be obtained by stacking the encoding output, finally a simple classifier can be learned from the transformed features. The deep reconstruction classification network (DRCN) [60] also learns a shared feature representation using autoencoder, and the output of the encoder is sent to two branches, that is, features of the source domain are sent directly to a classifier, while features of the target domain will continue to undergo a reconstruction process.

GAN can also be used to reconstruct the instances. The cycleGAN [61] uses two generators to learn two opposite mappings, $G : X \rightarrow Y$ and its inverse mapping $F : Y \rightarrow X$. This makes it possible to learn a common latent representation of two domains. Similar idea can also be found in dualGAN [62] which adds skip connections in both generators so as to share and leverage low-level features. The discoGAN [63] reformulates the domain-shift reducing problem as conditional image generation problem. Adopting the similar architecture of cycleGAN and dualGAN, discoGAN studies various distance functions to be used as loss function. After training, the model is able to change given properties while keeping the other contents unchanged.

V. Conclusion

In this survey, we make a retrospective study on robotic autonomous grasping techniques focusing on three points: grasp detection methods based on deep learning, object affordance detection methods, and model migration methods focusing on domain adaptation. We first introduce the basic grasping operation, then to improve the robot's understanding of environment and objects, review the methods of object affordance detection to achieve human-like grasp and complete various autonomous tasks, and finally review the model migration methods to apply the learned model in other scenes. This paper goes from basics to improving the intelligence of robots, step by step to bridge the gap between AI theory and practical applications. It shows readers the different research directions and progress from three aspects, so that readers can understand the field of autonomous robotic grasping faster and more comprehensively.

References

[1] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis—a survey," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 289–309, 2013.

[2] J. J. Gibson and L. Carmichael, *The senses considered as perceptual systems*. Houghton Mifflin Boston, 1966, vol. 2, no. 1.

[3] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic grasping of novel objects using vision," *The International Journal of Robotics Research*, vol. 27, no. 2, pp. 157–173, 2008.

[4] Y. Jiang, S. Moseson, and A. Saxena, "Efficient grasping from rgbd images: Learning using a new rectangle representation," in *2011 IEEE International conference on robotics and automation*. IEEE, 2011, pp. 3304–3311.

[5] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 705–724, 2015.

[6] J. Redmon and A. Angelova, "Real-time grasp detection using convolutional neural networks," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 1316–1322.

[7] F.-J. Chu, R. Xu, and P. A. Vela, "Real-world multiobject, multigrasp detection," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3355–3362, 2018.

[8] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours," in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 3406–3413.

[9] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," *arXiv preprint arXiv:1703.09312*, 2017.

[10] D. Morrison, P. Corke, and J. Leitner, "Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach," *arXiv preprint arXiv:1804.05172*, 2018.

[11] A. Zeng, S. Song, K.-T. Yu, E. Donlon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo *et al.*, "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 3750–3757.

[12] J. Cai, H. Cheng, Z. Zhang, and J. Su, "Metagrasp: Data efficient grasping by affordance interpreter network," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 4960–4966.

[13] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, "Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1521–1529.

[14] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," *arXiv preprint arXiv:1711.00199*, 2017.

[15] X. Deng, Y. Xiang, A. Mousavian, C. Eppner, T. Bretl, and D. Fox, "Self-supervised 6d object pose estimation for robot manipulation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 3665–3671.

[16] A. ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1455–1473, 2017.

[17] H. Liang, X. Ma, S. Li, M. Görner, S. Tang, B. Fang, F. Sun, and J. Zhang, "Pointnetgpd: Detecting grasp configurations from point sets," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 3629–3635.

[18] C. R. Qi, H. Su, K. Mo, and L. 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*, 2017, pp. 652–660.

[19] A. Mousavian, C. Eppner, and D. Fox, "6-dof graspnet: Variational grasp generation for object manipulation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2901–2910.

[20] Y. Qin, R. Chen, H. Zhu, M. Song, J. Xu, and H. Su, "S4g: Amodal single-view single-shot se (3) grasp detection in cluttered scenes," in *Conference on robot learning*. PMLR, 2020, pp. 53–65.

[21] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *arXiv preprint arXiv:1706.02413*, 2017.

[22] M. Hassanin, S. Khan, and M. Tahtali, "Visual affordance and function understanding: A survey," *arXiv preprint arXiv:1807.06775*, 2018.

[23] A. Nguyen, D. Kanoulas, D. G. Caldwell, and N. G. Tsagarakis, "Detecting object affordances with convolutional neural networks," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 2765–2770.

[24] A. Myers, C. L. Teo, C. Fermüller, and Y. Aloimonos, "Affordance detection of tool parts from geometric features," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 1374–1381.

[25] A. Nguyen, D. Kanoulas, D. G. Caldwell, and N. G. Tsagarakis, "Object-based affordances detection with convolutional neural networks and dense conditional random fields," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 5908–5915.

[26] T.-T. Do, A. Nguyen, and I. Reid, "Affordancenet: An end-to-end deep learning approach for object affordance detection," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 5882–5889.

[27] F.-J. Chu, R. Xu, and P. A. Vela, "Learning affordance segmentation for real-world robotic manipulation via synthetic images," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1140–1147, 2019.

[28] S. Deng, X. Xu, C. Wu, K. Chen, and K. Jia, "3d affordancenet: A benchmark for visual object affordance understanding," *arXiv preprint arXiv:2103.16397*, 2021.

[29] Y. Zhu, A. Fathi, and L. Fei-Fei, "Reasoning about object affordances in a knowledge base representation," in *European conference on computer vision*. Springer, 2014, pp. 408–424.

[30] P. Ardón, È. Pairet, R. P. Petrick, S. Ramamoorthy, and K. S. Lohan, "Learning grasp affordance reasoning through semantic relations," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4571–4578, 2019.

[31] C.-J. Kim, C. R. Nelson *et al.*, "State-space models with regime switching: classical and gibbs-sampling approaches with applications," *MIT Press Books*, vol. 1, 1999.

[32] H. Wu and G. S. Chirikjian, "Can i pour into it? robot imagining open containability affordance of previously unseen objects via physical simulations," *IEEE Robotics and Automation Letters*, vol. 6, no. 1, pp. 271–278, 2020.

[33] P. Abelha and F. Guerin, "Learning how a tool affords by simulating 3d models from the web," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 4923–4929.

[34] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[35] H. Wang, N. Wang, and D.-Y. Yeung, "Collaborative deep learning for recommender systems," in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 2015, pp. 1235–1244.

[36] V. Ramanishka, A. Das, D. H. Park, S. Venugopalan, L. A. Hendricks, M. Rohrbach, and K. Saenko, "Multimodal video description," in *Proceedings of the 24th ACM international conference on Multimedia*, 2016, pp. 1092–1096.

[37] I. Serban, A. Sordoni, Y. Bengio, A. Courville, and J. Pineau, "Building end-to-end dialogue systems using generative hierarchical neural network models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[38] K. Muandet, D. Balduzzi, and B. Schölkopf, "Domain generalization via invariant feature representation," in *International Conference on Machine Learning*. PMLR, 2013, pp. 10–18.

[39] J. Wang, C. Lan, C. Liu, Y. Ouyang, W. Zeng, and T. Qin, "Generalizing to unseen domains: A survey on domain generalization," *arXiv preprint arXiv:2103.03097*, 2021.

[40] A. Gretton, K. Borgwardt, M. J. Rasch, B. Scholkopf, and A. J. Smola, "A kernel method for the two-sample problem," *arXiv preprint arXiv:0805.2368*, 2008.

[41] B. Sun, J. Feng, and K. Saenko, "Correlation alignment for unsupervised domain adaptation," in *Domain Adaptation in Computer Vision Applications*. Springer, 2017, pp. 153–171.

[42] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[43] G. Kang, L. Jiang, Y. Yang, and A. G. Hauptmann, "Contrastive adaptation network for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4893–4902.

[44] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'14. Cambridge, MA, USA: MIT Press, 2014, p. 2672–2680.

[45] J. Huang, A. Gretton, K. Borgwardt, B. Schölkopf, and A. Smola, "Correcting sample selection bias by unlabeled data," *Advances in neural information processing systems*, vol. 19, pp. 601–608, 2006.

[46] M. Sugiyama, T. Suzuki, S. Nakajima, H. Kashima, P. von Bünau, and M. Kawanabe, "Direct importance estimation for covariate shift adaptation," *Annals of the Institute of Statistical Mathematics*, vol. 60, no. 4, pp. 699–746, 2008.

[47] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199–210, 2010.

[48] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 2066–2073.

[49] R. Gopalan, R. Li, and R. Chellappa, "Domain adaptation for object recognition: An unsupervised approach," in *2011 international conference on computer vision*. IEEE, 2011, pp. 999–1006.

[50] M. Zelikin, "Control theory and optimization i. encyclopedia of mathematical sciences, vol. 86," 2000.

[51] I.-H. Jhuo, D. Liu, D. Lee, and S.-F. Chang, "Robust visual domain adaptation with low-rank reconstruction," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 2168–2175.

[52] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," *arXiv preprint arXiv:1412.3474*, 2014.

[53] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *International conference on machine learning*. PMLR, 2015, pp. 97–105.

[54] A. Gretton, D. Sejdinovic, H. Strathmann, S. Balakrishnan, M. Pontil, K. Fukumizu, and B. K. Sriperumbudur, "Optimal kernel choice for large-scale two-sample tests," in *Advances in neural information processing systems*. Citeseer, 2012, pp. 1205–1213.

[55] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International conference on machine learning*. PMLR, 2015, pp. 1180–1189.

[56] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7167–7176.

[57] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4068–4076.

[58] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 1096–1103.

[59] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, and L. Bottou, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion." *Journal of machine learning research*, vol. 11, no. 12, 2010.

[60] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li, "Deep reconstruction-classification networks for unsupervised domain adaptation," in *European Conference on Computer Vision*. Springer, 2016, pp. 597–613.

[61] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[62] Z. Yi, H. Zhang, P. Tan, and M. Gong, "Dualgan: Unsupervised dual learning for image-to-image translation," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2849–2857.

[63] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, “Learning to discover cross-domain relations with generative adversarial networks,” in *International Conference on Machine Learning*. PMLR, 2017, pp. 1857–1865.