2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

Using Synthetic Data and Deep Networks to Recognize Primitive Shapes for Object Grasping

Yunzhi Lin[1]*, Chao Tang[1]*, Fu-Jen Chu[1], and Patricio A. Vela [1]

Abstract— **A segmentation-based architecture is proposed to decompose objects into multiple primitive shapes from monocular depth input for robotic manipulation. The backbone deep network is trained on synthetic data with 6 classes of primitive shapes generated by a simulation engine. Each primitive shape is designed with parametrized grasp families, permitting the pipeline to identify multiple grasp candidates per shape primitive region. The grasps are priority ordered via proposed ranking algorithm, with the first feasible one chosen for execution. On task-free grasping of individual objects, the method achieves a 94% success rate. On task-oriented grasping, it achieves a 76% success rate. Overall, the method supports the hypothesis that shape primitives can support task-free and task-relevant grasp prediction.**

I. Introduction

Manipulation is a multi-step task consisting of sequential actions applied to an object, including: perception, path planning, and closure of the gripper, followed by a task-relevant motion with the grasped object [1]. Due to the diversity of objects that a robotic arm could grasp, the grasping process remains an open problem in the field of robotics. Analytical or model-based approaches have trouble addressing this diversity. Recent research has turned to deep learning as a robust means to score or detect grasps.

Deep learning is a data-driven approach requiring large datasets to recover a desired input/output function, which for grasping is often an image/grasp pair (allowing multiple grasps per image). Manually annotated datasets tend to be limited in volume [2], leading roboticists to exploit robotic simulators [3], [4] or actual deployment [5] for automatic data generation. Many of these emphasize the grasp as opposed to the object, with a good reason as object-centric approaches would require creating 3D models or scanning real objects in large volume [6], [7], and concomitantly require a high accuracy detector. Instead, deep networks aim to generalize across object classes to establish general grasping rules from visual evidence. It could implicitly learn the notion of shape within its internal feature representation.

This paper proposes to complement that idea with a more geometric and explicitly shape-centric approach based on the notion of primitive shapes. Primitive shapes offer a powerful means to alleviate the data insufficiency problem [8], [9], [10] by abstracting target objects to primitive shapes with *a priori* known grasp configurations. Most primitive shape methods represent objects as a single shape from a small library [9], [11] or apply model-based rules to deconstruct objects [8]. As a result, they do not handle novel shapes with unmodeled geometry or being the union of primitive shapes.

Building from Yamanobe *et al.*'s model decomposition idea [8], this paper aims to permit a primitive shapes detector to generalize its grasp strategies to household objects. The detector exploits the state-of-the-art instance segmentation deep network Mask R-CNN [12] trained to segment a depth image according to the primitive shapes it contains. Synthetic ground truth data based on parametrized sets of primitive shape classes and their grasp families avoids extensive manual annotation. The shape classes are sufficiently representative of object parts associated with household objects yet low enough in cardinality that grasp family modeling is quick. Domain randomization [10], [13] over the parametrized shapes generates a rich set of ground truth input/output data based on a robotics simulation engine [14]. The result is a large synthetic dataset composed of different primitive shapes combinations, quantities, and layouts.

Modern robotics simulation engines [14], [15] render and simulate virtual environments quite cleanly, often leading to sensor imagery with fewer defects than real sensors. Depth sensor like the Kinect v1 loses details and introduces noise during the depth capture [16], [17], leading to a distribution shift or gap between the simulated data and depth sensor output. The gap is addressed in a bi-directional manner by lightly corrupting the simulated data and denoising the depth image data. The intent is to introduce real sensor artifacts that cannot be removed in the simulated images, and to denoise the real images to match the simulated images.

Employing the deep network for grasping enables a robotic arm with depth camera to identify visible primitive shapes, and to recover grasp options for an object from the shapes. To arrive at the overall system, from training to deployment, the paper covers the following contributions, which are experimentally tested and compared to published baselines:

- An automated ground truth generation strategy to rapidly generate input/output data. It uses 6 classes of primitive shapes with parametrized grasp families.
- A deep network, segmentation-based pipeline first decomposing objects into multiple primitive shapes from a depth image, followed by a candidate grasp selection pipeline.

II. Related Work

Grasping is a mechanical process which can be described mathematically given prior knowledge of the target object's properties (geometry, hardness, etc.), the hand contact model,

[1] Y. Lin, C. Tang, F. Chu, and P.A. Vela are with the School of Electrical and Computer Engineering, and the Institute for Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, GA. {ylin466, ctang84, fujenchu, pvela}@gatech.edu
This work was supported in part by NSF Award #1605228
* The first two authors contributed equally.

and the hand dynamics. Mechanics-based approaches with analytical solutions work well for some objects but cannot successfully apply to other, often novel, targets [18], [19], [20]. With advances in machine learning, these methods gave way to purely data-driven approaches [21] or combined approaches employing analytic scoring with *image-to-grasp* learning [3]. Contemporary solutions employ deep learning [22] and leverage available training data.

Deep learning strategies primarily take one of three types. The first type exploits the strong detection or classification capabilities of deep networks to recognize candidate structured grasping representations [23], [24], [25], [26]. The most common representation is the $SE(2) \times \mathbb{R}^2$ grasp representation associated with a parallel plate gripper. As a computer vision problem, recognition accuracy is high (up to around 95%), with a performance drop during robotic implementation (to around 90% or less). Training involves image/grasp datasets obtained from manual annotation [27] or simulated grasping [4], [26]. Within this category, there is also a mixed approach, DexNet [3], using random sampling and analytical scoring followed by deep network regression to output refined, learned grasp quality scores for grasp selection. By using simulation with an imitation learning methodology, tens of thousands to millions of annotations support DexNet regression training. Success rates vary from 80% to 93% depending on the task. When sufficient resources are available, the second strategy replaces simulation with actual experiential data coupled to deep network reinforcement learning methods [5], [28]. Often, methods based on simulation or experience are configuration-dependent; they learn for specific robot and camera setups. The third strategy is based on object detection or recognition [21]. Recent work employed deep learning to detect objects and relative poses to inform grasp planning [29], while another learnt to perform object-agnostic scene segmentation to differentiate objects [30] and aided DexNet grasp selection process. Like [30], this paper focuses on where to find candidate grasps as opposed to quality scoring candidate grasps.

Deep learning grasp methods suffer from two related problems, sparse grasp annotations or insufficiently rich data (i.e., covariate shift). The former can be seen in Figure 1, which shows an image from the Cornell dataset [2] and another from the Jacquard dataset [4]. Both lack annotations in graspable regions due to missing manual annotation or a false negative in the simulated scenario (either due to poor sampling or incorrect physics). Sampling insufficiency can be seen in [26], where the DexNet training policy was augmented with an improved (on-policy) oracle providing a richer sampling space. Yet, sampling from a continuous space is bound to under-represent the space of possible options, especially as the dimension of the parametric grasp space increases. This paper proposes to more fully consider shape primitives [31] due to their known, parametrized grasp families [8], [32]. The parameterized families provide a continuum of grasp options rather than a sparse sampling. A complex object can be decomposed into parts representing distinct surface categories based on established primitives.

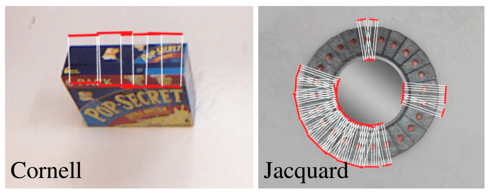

Fig. 1. Grasp annotation data with missing grasp candidates.

Deep network approaches for shape primitive segmentation to inform grasping do not appear to be well studied. Past research explored shape primitive approaches in the context of traditional point cloud processing and fitting for the cases of superquadric [33] and box surfaces [34]. Approaches were also proposed to simply model each object as a single primitive shape [9], [35], which did not exploit the potential of primitive shapes to generalize to unseen/novel objects.

III. Grasping from Primitive Shapes Recognition

This paper explores the potential value of using deep networks to segment a scene according to the surface primitives contained within it, in order to establish *how* one may grasp. Once the object or region to grasp is known, post-processing recovers the shape geometry and the grasp family associated with the shape. The state-of-the-art instance segmentation deep network Mask R-CNN [12] serves as the backbone network for converting depth images into segmented primitive shape images. Importantly, a synthetically generated training set using only shape primitives in concert with domain randomization [10] covers a large set of scene visualizations. The ability to decompose unseen objects into distinct shape regions, often with explicitly distinct manipulation affordances, permits task-oriented grasping [35].

The vision-based robotic grasping problem here presumes the existence of a depth image $\mathcal{D} \in \mathbb{R}^{H \times W}$ (H and W are image height and width) capturing a scene containing an object to grab. The objective is to abstract the scene into a set of primitive shapes and generate grasp configurations from them. A complete solution involves establishing a routine or process, f, mapping the depth image $\mathcal{D}$ to a grasp $\mathcal{G} = f(\mathcal{D}) \in SE(3)$. The grasp configuration $\mathcal{G} \in SE(3)$ specifies the final pose in the world frame of the end-effector.

Per Figure 2, the process is divided into three stages. In the first stage, the depth image $\mathcal{D}$ gets segmented according to defined primitive shape categories indexed by the set I. The primitive shape segmentation images are $\mathcal{P}_i$ for $i \in I$. The segmentation $\mathcal{P}_i$ and the depth image $\mathcal{D}$ generate segmented point clouds in 3D space for the primitive surfaces attached to the label i. In the second stage, when the grasp target is established, the surface primitives attached to the target grasp region are converted into a corresponding set of primitive shapes P_j in 3D space, where j indexes the different surface primitive segments. In the third stage, the parametrized grasp families of the surface primitives are used to generate grasp configurations $\mathcal{G}_\alpha$ for $\alpha \in \mathbb{N}^+$. A prioritization process leads to rank-ordered grasps with the first feasible grasp being the one to execute. This section details the three stages and the deep network training method.

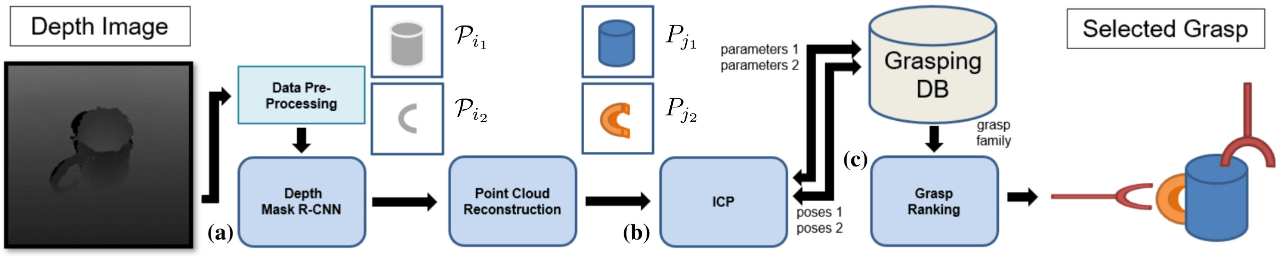

Fig. 2. The proposed deep network, segmentation-based pipeline. From monocular depth input, objects are segmented into primitive shape classes, with the object to grasp extracted and converted into primitive shape point clouds. Surface model-fitting, grasp scoring, and grasp selection processes follow. **(a)** Depth input is segmented into primitive shapes; **(b)** The best matching shape and pose per primitive shape is identified; **(c)** Candidate grasps are priority-ranked and tested for feasibility with the first feasible grasp chosen for physical robot executions.

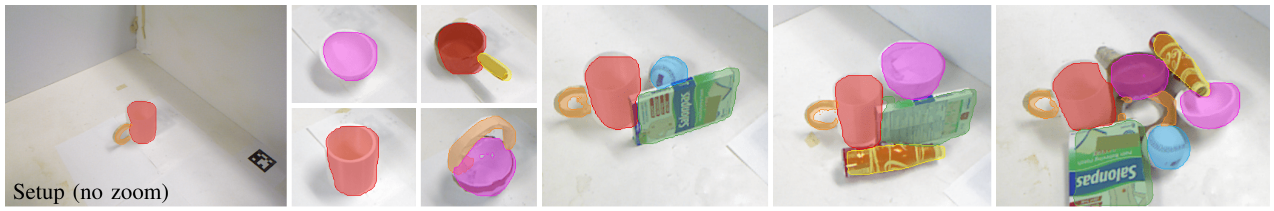

Fig. 3. Sample segmentation outcomes for test scenarios consisting of individual and multiple objects (zoomed and cropped images).

TABLE I
PRIMITIVE SHAPE CLASSES

Class	Parameters	Shape Template (unit: cm)
Cylinder (wide, tall)	(r_{in}, r_{out}, h)	$\sigma \in [2.2, 3.0]$ $\rho_{wide} = (1, 1.15, 2.45)$
		$\sigma \in [2, 6.5]$ $\rho_{tall} = (1, 1.15, 5.063)$
Ring	(r_{in}, r_{out}, h)	$\sigma \in [1.85, 5.0]$ $\rho = (1, 1.25, 0.6125)$
Cuboid (wide, tall)	(h, w, d)	$\sigma \in [2.2, 3.0]$ $\rho_{wide} = (6.25, 4, 11)$ $\rho_{tall} = (18.49, 1.9, 12.09)$
Stick	(r, l)	$\sigma \in [5.5, 19]$, $\rho = (1.25, 1)$
Semi-sphere	(r_{in}, r_{out})	$\sigma \in [3.5, 8]$, $\rho = (0.9, 1)$
Sphere	r	$\sigma \in [2.25, 4.5]$, $r = 1$

r - radius, r_{in} - inner radius, r_{out} - outer radius
h - height, w - width, d - depth, l - length

A. Primitive Shape Segmentation Using Mask R-CNN

The proposed approach hypothesizes that commonly seen household objects can be decomposed into one or more primitive shapes and represented by pre-defined shape parameters. After studying several household object datasets [7], [36], [37], [38], the set of primitive shapes was decided to be: *cylinder*, *cuboid*, *ring*, *sphere*, *semi-sphere*, and *stick*.

Primitive shapes training data may be automatically synthesized through gridded sampling within the parametric domain of each class, as specified in Table I. Sections IV-A and III-D detail the training method used to synthetically generate depth images and known segmentations from the parametric shape classes. The trained Mask-RCNN [12] network converts an input depth image into segmentations reflecting hypothesized primitive shapes, as shown in Figure 2(a). Segmentations for different input depth images overlaid on the corresponding, cropped RGB images are found in Figure 3 (color coding is red: *cylinder*, orange: *ring*, green: *cuboid*, yellow: *stick*, purple: *semi-sphere*, and blue: *sphere*). The segmentation captures object's primitive shapes.

B. Shape Extraction and Estimation, and Grasp Family

Given a target object to grasp, the intersecting shape primitive regions are collected and converted into separate partial point clouds. Each point cloud needs to be associated with a parametric model of the known shape. Using an object and grasp database of examplar shapes, a multi-start Iterative Closet Point (ICP) algorithm matches the exemplar shapes to the point cloud. Multiple starting points reduce local minima issues. The final match with the lowest error score is selected as the object model for hypothesizing grasps. In addition to the model, the transformation aligning the point cloud with the shape is kept for mapping grasps to the world frame. This matching step is depicted in Figure 2(b). Given the identified type of shape and its corresponding shape parameters, one or more families of grasps are recovered. The geometric primitives do not need to match the target region exactly, as long as the error between the object and the best matching primitive shapes is small relative to the grasp policy.

Each object class has a family of grasps based on its geometry, with each member of the family reflecting grasps associated with transformation by a group action due to the symmetry of the primitive. For grabbing wide cylinders, there are two members corresponding to grabbing from the top or from the bottom, with the free parameter being rotation about the cylinder axis. For grabbing a thin cylinder, there is one two-parameter set corresponding to translation along or rotation about the cylinder axis. Example grasps from these described families are depicted in Figure 4. Under ideal conditions (i.e., no occlusion, no collision, and reachable) all grasps are possible (e.g., the object is floating). In reality, only a subset of all possible grasps will be feasible. Using the shape primitive grasp family avoids the problem of incomplete annotations or sparse grasp sampling.

C. Grasp Prioritization and Selection

The final step prior to execution is to select one grasp from the set of candidate grasps. A simple geometric grasp

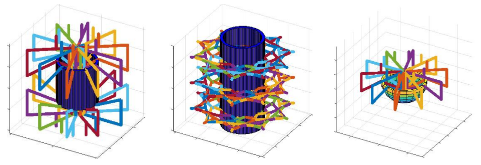

Fig. 4. Grasp family for wide cylinder, tall cylinder, and semi-sphere.

prioritization scoring function is adopted, based on the required hand pose relative to the manipulator base frame, $\mathcal{G} = \mathcal{G}^{\mathcal{B}}_{\mathcal{H}}$. The prioritization scheme prefers the desired grasp to minimize translation and favor approaching from above.

The grasp prioritization score consists of contributions from the translational and rotational components of a test grasp pose. The translation score contribution depends on the length of the translation element of $\mathcal{G}$. Define the translation cost to be $C_T(T) = \||T\||$, where T is the translation interpreted to be a vector in $\mathbb{R}^3$. The rotational contribution regards the equivalent quaternion $\mathfrak{q} \in SO(3)$ as a vector in $\mathbb{R}^4$ and applies the following positive, scalar binary operation to obtain the orientation grasp cost:

$$C_R(\mathfrak{q}) = \vec{w} \odot \vec{\mathfrak{q}}, \quad \text{where } \vec{w} = (0, \omega_1, \omega_2, \omega_3)^T, \ \omega_i > 0, \quad (1)$$

with $\vec{a} \odot \vec{b} \equiv |a^1 b^1| + |a^2 b^2| + |a^3 b^3| + |a^4 b^4|$. This cost prioritizes vertical grasps by penalizing grasps that do not point up/down. Alternative weightings are possible depending on the given task or the robot configuration. The costs are computed for all grasp candidates, then converted into a score by normalizing them over the range of obtained scores,

$$\begin{aligned} s_R(C_R) &= 1 - (C_R - C_R^{min})/(C_R^{max} - C_R^{min}) \\ s_T(C_T) &= 1 - (C_T - C_T^{min})/(C_T^{max} - C_T^{min}) \end{aligned} \quad (2)$$

where the $\cdot^{min}$ and $\cdot^{max}$ superscripts denote the min and max over all scores grasps. Multi-score ranking is applied by taking s_R and s_T into a final grasp prioritization score.

$$\gamma_{MC}(s_R, s_T) = \lambda_R s_R + \lambda_T s_T \quad \text{for } \lambda_R, \lambda_T > 0. \quad (3)$$

After ordering the grasps according to their prioritization score, the actual grasp applied is the first one to be feasible when a grasp plan is made from the current end-effector pose to the target grasp pose. This third and final step in the grasp identification process is shown Figure 2(c).

D. Domain Alignment between Simulation and Reality

Simulators [14], [15] support data generation by automating data collection in virtual environments, but do so using idealized physics or sensing. Some physical effects are too burdensome to model. To alleviate this problem, the images from both sources, the simulation and the depth sensor, are modified to better match. The objective is to minimize the corrections applied, therefore the first step is to reduce or eliminate the sources of discrepancies by configuring both environments to match. This includes the camera's intrinsic and extrinsic parameters, and the background scene. Comparing images from both sources, the main gap remaining is the sensing noise introduced by the low-fidelity Kinect v1 depth sensor [16], [17]. The Kinect has occlusion artifacts arising from the baseline between the active illuminator and the

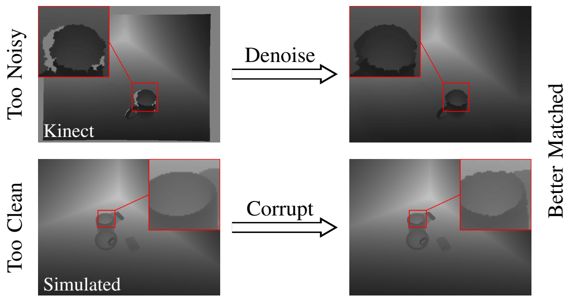

Fig. 5. Bi-directional image filtering to align training data and real data. An oil painting filter applied to training imagery simulates the noise of the Kinect depth sensor. Temporal averaging and spatial median filtering regularize the Kinect depth image during run-time.

imaging sensor, plus from measurement noise. The denoising process applies temporal averaging, boundary cropping, then median filtering. Once the Kinect depth imagery is denoised, the next step is to corrupt the simulated depth imagery to better match the visual characteristics of the Kinect. The primary source of uncertainty is at the depth edges or object boundaries due to the properties of the illuminator/sensor combination. The simulated imagery should be corrupted at these same locations. The simulated environment has both a color image and a depth image. The color image is designed to provide both the shape primitive label and the object ID, thereby permitting the extraction of object-wise boundaries. After finding the object boundary pixels, they are dilated to obtain enlarged object boundaries regions, then an oil painting filter [39], [40] corrupts the depth data of these regions. Considering that manipulation is only possible within the manipulator's workspace, the depth values from both sources were clipped and scaled to map to a common interval. Figure 5 depicts this bi-directional process showing how it improves alignment between the two sources.

IV. Training, Experiments, and Evaluation

This section covers automated ground truth synthesis and training, the robot setup, and evaluation criteria.

A. Dataset Generation

Based on the hypothesis that a dataset with diverse combinations of primitive shapes could induce learning generalizable to household objects, the dataset generation procedure consists of the following degrees of freedom (1) primitive shape parameter; (2) placement order; (3) initial $SE(3)$ pose assigned; and (4) mode of placement (there are three modes: *free-fall*, *straight up from the table-top*, and *floating in the air*). The shape primitives from Table I are reduced to one parameter families after analyzing the household object databases. Each has a fixed parameter vector ρ and scaling factor σ, as described in the third column of Table I. Both cylinders and cuboids are modeled to have a *wide-and-short* category, and a *tall-and-thin* category (denoted *wide* and *tall* base on the largest parameter) by defining a default parameter set for each category. The scaling factor σ defines the one parameter family per class. The *ring*, *semi-sphere*, and *sphere* vectors are scaled for all coordinates. The *stick* category is scaled for the length coordinate l. The process samples a sufficiently rich set of visualized shapes once self-occlusion and object-object occlusion effects are factored in.

For creating instances of the world, the scaling factor is discretized into 10 steps. Every image has one instance of each primitive class uniformly selected. The insertion order of the 6 objects is random. The initial poses are uniformly randomly determined within a bounding volume. The placement type is uniformly randomly determined. A total of 100,000 scenes of different primitive shapes combinations are generated. For each instance, RGB and depth images are collected. Shape color coding provides segmentation ground truth and primitive shape ID.

B. Data Preprocessing and Training

The simulated depth images are processed then corrupted by a region-specific oil-painting filter (§III-D). To be aligned with the Mask R-CNN input, the single depth channel is duplicated across the three input channels. The backbone, ResNet-50-FPN, is trained from scratch on corrupted depth images (PyTorch 1.0) for 100,000 iterations with 4 images per mini-batch. The primitive shape dataset is divided into a 75%/25% training/testing split. The learning rate is set to 0.01 and divided by 10 at iterations 25000, 40000, and 80000. The workstation has a single NVIDIA 1080Ti (Pascal arch.) with cudnn-7.5 and cuda-9.0. Dataset generation takes 72 hours, and training takes 24 hours (4 days in total).

C. Robotic Arm Experimental Setup and Parameters

The eye-to-hand robotic arm and RGB-D camera setup used for evaluation is shown in Figure 6 (left). The camera to manipulator base frame calibration uses an ArUco tag captured by the camera. A set of 3D printed shapes is used for *known objects* testing, Figure 6 (top right). An additional set of objects is used for *novel objects* testing, Figure 6 (bottom right).

The primitive shape and grasp family database is populated by selecting 10 exemplars from each primitive shape type. Each continuous grasp set is likewise discretized according to the dimensions of our gripper so that neighboring grasps are not too similar. The scoring weights in (3) are set equal, $\lambda_R = \lambda_T = 0.5$. Collision checking for grasp feasibility is done by the *Planning Scene* module in *MoveIt!* [41]. Open-loop execution is performed with the plan of the top grasp.

D. Evaluation Metrics

Evaluation of the proposed pipeline consists of testing on novel input data as a visual recognition problem and testing on the experimental system. For visual segmentation evaluation, segmentation accuracy is computed by F^{ω}_{β} [42]:

$$F^{\omega}_{\beta} = (1+\beta^2)\, Pr^{\omega} \cdot Rc^{\omega} \Big/ (\beta^2 \cdot Pr^{\omega} + Rc^{\omega}), \qquad (4)$$

where Pr is precision and Rc is recall, $\beta = 1$, and ω is a Gaussian smoothing factor. For the robotic arm, the success rate (percentage) is recorded. Each experiment configuration consists of 10 repeated runs. Success occurs when the target object is grasped, lifted, and held for at least 10 seconds.

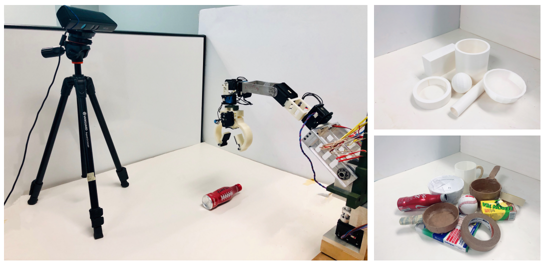
Fig. 6. Experimental setup and images of known and novel object sets.

TABLE II
PERFORMANCE ON 3D PRINTED PRIMITIVE SHAPES

	Original	Corrupted		Original	Corrupted
Cylinder	0.918	0.917	Ring	0.903	0.904
Cuboid	0.822	0.824	Stick	0.787	0.838
Semi-sphere	0.919	0.913	Sphere	0.661	0.835
			All	0.835	0.872

V. RESULTS

This section details the experiments performed and their outcomes. Manipulation tests include an in-class grasping test consisting of individual objects whose shape matches one of the primitive shape classes, and an out-of-class grasping test consisting of novel objects, some of which composed of multiple shape classes. There is also a task-oriented grasping task where a specific primitive shape must be grasped, as would be done when performing a specific task with the object. The fourth is a stress test of the system: a multi-object clearing task. The baseline implementations are [9], a primitive-shape-based system for household objects, and [25] a publicly available deep network RGB-D grasping approach. The implementation [9] was reproduced, with the primitives being: spherical, cylindrical, and box-like. Each object is approximated by the best fitting primitive shape.

A. Vision

Segmentation tests are performed on a set of 3D printed primitive shapes and compared to manual segmentations. The tested implementations include a network trained with the original simulated images (no corruption) and with the oil-painting corrupted images. Per Table II , the F^{ω}_{β} segmentation accuracy improved from 0.835 to 0.872 (ranges over $[0, 1]$), with the primary improvement sources being for the *sphere* shape class followed by the *stick class*. The segmentation accuracy is sufficient to capture and label significant portions of an object's graspable shape regions, see Figure 3.

B. Physical Grasping of Known and Unknown Objects

These tests evaluate the ability of the pipeline to pick up an individual object in the absence of obstacles or clutter. The *known objects* grasping test outcomes are in Table III . The proposed primitive shape Mask R-CNN (PS-CNN) approach outperformed the baseline methods. Trained purely on primitive synthetic data, the proposed approach bridges the gap between simulation and real-world application. The baseline [9] achieved a competitive success rate on shapes matching the spherical, cylindrical, and box-like primitives. The baseline [25] had the lowest success rate.

TABLE III

Primitive Shapes Grasping (Known)

	PS-CNN	[9]	[25]
Cuboid	8	8	5
Cylinder	10	9	8
Semi-sphere	9	5	7
Stick	10	7	6
Ring	9	6	5
Sphere	9	8	8
Success (%)	91.7	71.7	65.0

TABLE IV

Real Objects Grasping (Unkown)

	PS-CNN	[9]	[25]
Bowl	10	6	7
Mug	10	6	8
Box	10	9	9
Baseball	8	8	8
Tape	9	9	8
Bottle	9	7	8
Success (%)	93.3	75.0	80.0

TABLE V

Task-Oriented and Task-free Grasp

Target Objects	Task-Free		Task-Oriented	
	[9]	Ours	[9]	Ours
Mug	6/10	10/10	-	8/10
Pot	3/10	9/10	-	7/10
Pan	5/10	9/10	-	7/10
Basket	6/10	10/10	-	8/10
Handbag	9/10	10/10	-	8/10
Success Rate (%)	58.0	96.0	0	76.0

TABLE VI

Grasping Comparison from Published Works

Approach	Year	Settings		Success Rate (%)
		Objects	Trials	
Lenz *et al.*[27]	2015	30	100	84/89*
Pinto *et al.* [43]	2016	15	150	66
Watson *et al.* [44]	2017	10	-	62
DexNet 2.0 [3]	2017	10	50	80
Lu *et al.* [45]	2018	10	-	84
DexNet 4.0 [46]	2019	50	5	50
Satish *et al.* [26]	2019	8	80	87.5
PS-CNN	-	10**	100	94

* Success rate of 84%/89% achieved on Baxter/PR2 robot.
** Combines the unique objects from Tables IV and V.

TABLE VII

Multi-Object Grasping Comparison

Method	#Obj.	#Sel.	#Trials	TC	S	OC	C
[43]	21	10	5	None	38	100	100
[47]	10	10	15	3Seq	84	77	–
[3]	25	5	20	5Seq	92	100	100
[5]	25	25	4	31G	82.1	99	75
[48]	16	8	15	+2	86.1	87.5	–
[46]	25	5	20	5Seq	94	100	100
PS-CNN	10	3-5	10	+2	50	80	70

The *unknown objects* grasp test outcomes are in Table IV. The PS-CNN obtains the best performance. The baseline methods also showed good performance, but the success rate continued to be lower than the proposed approach.

Table VI collects success rate and testing details for other state-of-the-art approaches employing only a single gripper (no suction-based end-effector). Even though the test objects may differ from ours, the general object classes are similar. Organized sequentially, the top performing results are the earliest (Lenz *et al.*) and the latest (Satish *et al.*) approaches, achieving below 90% over up to 100 trials. The proposed approach has a higher success rate suggesting that identifying primitive shape regions may aid grasping.

C. Task-Oriented Grasping on Real Objects

The value of segmenting objects is that the distinct shape regions may correspond to task-based grasp preferences. The earlier tests focused on task-free grasping (or simply pick-n-place operations). Here, each connected primitive shape region is presumed to correspond to a specific functional part. Task-oriented grasp testing is performed, where a specific region must be grasped. Each object consists of more than one functional part. Only [9] was tested to serve as a baseline since it is shape-based. Table V reports the performance of the systems on the grasping tests. The method [9] did not perform as well as in prior tests since it applies a single shape model only, and cannot model more complex shape profiles. The PS-CNN did well but experienced a 20% drop in the success rate when limited to a specific object part. Additional study should be placed on precision grasping of specific object parts in addition to general grasping. The task-oriented approach scores comparably to some of the outcomes of Table VI, suggesting that strong task-oriented performance should be possible.

D. Multi-Object Grasping on Real Objects

The current study is aimed at establishing grasp candidates for individual objects rather than clutter removal or bin-picking, however additional testing was performed to gauge the limits of the implementation. Performance is not expected to be high, since the current version would be equivalent to DexNet 1.0 or 2.0 (vs DexNet 2.1). From the set of 10 objects, a smaller set of 3 to 5 objects are randomly selected and placed on the workspace, see Figure 3. Following [48], the robot has $n+2$ grasp attempts to remove the objects ([5] used $n+5$ for $n = 25$). We calculated the grasp success percentage (S), the object clearance percentage (OC), and the completion percentage (C). Table VII compares these statistics with other published works. There are some differences regarding trial termination criteria (TC), with kG signifying up to k grasp attempts, $+k$ signifying $n+k$ allowed grasps for n objects, and kSeq meaning k sequential failures. Combining the PS-CNN with a downstream grasp quality CNN (GQ-CNN) trained to recognize grasps from a richer set of camera perspectives might improve object grasping in clutter. So would closed-loop operation to correct the tracking error of the low-cost servomotors in the arm.

VI. Conclusion

This paper leverages advances in deep learning to realize a shape primitive segmentation approach to grasping. Shape primitive knowledge permits grasp recovery from known grasp families. It returns to a classical paradigm for grasping and shows that high performance grasp candidates can be learnt from simulated visual data without simulating grasp attempts. The segmentation-based approach permits task-oriented object grasping in contrast to approaches emphasizing grasp learning. Robotic grasping experiments indicate a 94% grasp success rate for task-free grasping and 76% for task-oriented learning. Future work aims to improve cluttered grasping success by using contemporary grasp networks to score the grasp candidates for success likelihood.

References

[1] M. Ciocarlie, K. Hsiao, E. G. Jones, S. Chitta, R. B. Rusu, and I. A. Şucan, "Towards reliable grasping and manipulation in household environments," in *Experimental Robotics*, 2014, pp. 241–252.

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

[3] 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," in *Robotics: Science and Systems*, 2017.

[4] A. Depierre, E. Dellandréa, and L. Chen, "Jacquard: A large scale dataset for robotic grasp detection," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018, pp. 3511–3516.

[5] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 421–436, 2018.

[6] W. Wohlkinger, A. Aldoma, R. B. Rusu, and M. Vincze, "3dnet: Large-scale object class recognition from cad models," in *IEEE International Conference on Robotics and Automation*, 2012, pp. 5384–5391.

[7] B. Calli, A. Singh, J. Bruce, A. Walsman, K. Konolige, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Yale-cmu-berkeley dataset for robotic manipulation research," *The International Journal of Robotics Research*, vol. 36, no. 3, pp. 261–268, 2017.

[8] N. Yamanobe and K. Nagata, "Grasp planning for everyday objects based on primitive shape representation for parallel jaw grippers," in *IEEE International Conference on Robotics and Biomimetics*, 2010, pp. 1565–1570.

[9] S. Jain and B. Argall, "Grasp detection for assistive robotic manipulation," in *IEEE International Conference on Robotics and Automation*, 2016, pp. 2015–2021.

[10] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017, pp. 23–30.

[11] C. Eppner and O. Brock, "Grasping unknown objects by exploiting shape adaptability and environmental constraints," in *IROS*, 2013, pp. 4000–4006.

[12] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *IEEE International Conference on Computer Vision*, 2017, pp. 2961–2969.

[13] J. Tobin, L. Biewald, R. Duan, M. Andrychowicz, A. Handa, V. Kumar, B. McGrew, A. Ray, J. Schneider, P. Welinder *et al.*, "Domain randomization and generative models for robotic grasping," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018, pp. 3482–3489.

[14] E. Rohmer, S. P. Singh, and M. Freese, "V-rep: A versatile and scalable robot simulation framework," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 1321–1326.

[15] B. O. Community, *Blender - a 3D modelling and rendering package*, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.

[16] B. Planche, Z. Wu, K. Ma, S. Sun, S. Kluckner, O. Lehmann, T. Chen, A. Hutter, S. Zakharov, H. Kosch *et al.*, "Depthsynth: Real-time realistic synthetic data generation from cad models for 2.5 d recognition," in *IEEE International Conference on 3D Vision*, 2017, pp. 1–10.

[17] C. Sweeney, G. Izatt, and R. Tedrake, "A supervised approach to predicting noise in depth images," in *IEEE International Conference on Robotics and Automation*, 2019, pp. 796–802.

[18] C.-P. Tung and A. C. Kak, "Fast construction of force-closure grasps," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 615–626, 1996.

[19] D. Prattichizzo, M. Malvezzi, M. Gabiccini, and A. Bicchi, "On the manipulability ellipsoids of underactuated robotic hands with compliance," *Robotics and Autonomous Systems*, vol. 60, no. 3, pp. 337–346, 2012.

[20] C. Rosales, R. Suárez, M. Gabiccini, and A. Bicchi, "On the synthesis of feasible and prehensile robotic grasps," in *IEEE International Conference on Robotics and Automation*, 2012, pp. 550–556.

[21] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesisa survey," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 289–309, 2014.

[22] S. Caldera, A. Rassau, and D. Chai, "Review of deep learning methods in robotic grasp detection," *Multimodal Technologies and Interaction*, vol. 2, no. 3, p. 57, 2018.

[23] J. Watson, J. Hughes, and F. Iida, "Real-world, real-time robotic grasping with convolutional neural networks," in *Annual Conference Towards Autonomous Robotic Systems*, 2017, pp. 617–626.

[24] D. Park and S. Y. Chun, "Classification based grasp detection using spatial transformer network," in *arXiv preprint arXiv:1803.01356*, 2018.

[25] F. 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.

[26] V. Satish, J. Mahler, and K. Goldberg, "On-policy dataset synthesis for learning robot grasping policies using fully convolutional deep networks," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1357–1364, 2019.

[27] 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.

[28] A. Zeng, S. Song, S. Welker, J. Lee, A. Rodriguez, and T. Funkhouser, "Learning synergies between pushing and grasping with self-supervised deep reinforcement learning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018, pp. 4238–4245.

[29] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, "Deep object pose estimation for semantic robotic grasping of household objects," in *Conference on Robot Learning*, 2018.

[30] M. Danielczuk, M. Matl, S. Gupta, A. Li, A. Lee, J. Mahler, and K. Goldberg, "Segmenting unknown 3d objects from real depth images using mask r-cnn trained on synthetic data," in *IEEE International Conference on Robotics and Automation*, 2019, pp. 7283–7290.

[31] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic grasp planning using shape primitives," in *IEEE International Conference on Robotics and Automation*, 2003, pp. 1824–1829.

[32] Y. Shiraki, K. Nagata, N. Yamanobe, A. Nakamura, K. Harada, D. Sato, and D. N. Nenchev, "Modeling of everyday objects for semantic grasp," in *IEEE International Symposium on Robot and Human Interactive Communication*, 2014, pp. 750–755.

[33] C. Goldfeder, P. K. Allen, C. Lackner, and R. Pelossof, "Grasp planning via decomposition trees," *IEEE International Conference on Robotics and Automation*, 2007.

[34] K. Huebner and D. Kragic, "Selection of robot pre-grasps using box-based shape approximation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 1765–1770.

[35] K. Fang, Y. Zhu, A. Garg, A. Kuryenkov, V. Mehta, L. Fei-Fei, and S. Savarese, "Learning task-oriented grasping for tool manipulation from simulated self-supervision," in *Robotics: Science and Systems*, 2018.

[36] D.-Y. Chen, X.-P. Tian, Y.-T. Shen, and M. Ouhyoung, "On visual similarity based 3d model retrieval," in *Computer graphics forum*, vol. 22, no. 3, 2003, pp. 223–232.

[37] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, and D. Jacobs, "A search engine for 3d models," *ACM Transactions on Graphics*, vol. 22, no. 1, pp. 83–105, 2003.

[38] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser, "The princeton shape benchmark," in *Shape Modeling Applications*, 2004, pp. 167–178.

[39] A. C. Sparavigna and R. Marazzato, "Cld-shaped brushstrokes in non-photorealistic rendering," in *arXiv preprint arXiv:1002.4317*, 2010.

[40] S. Mukherjee, "Study on performance improvement of oil paint image filter algorithm using parallel pattern library," *Computer Science & Information Technology*, p. 39, 2014.

[41] S. Chitta, I. Sucan, and S. Cousins, "Moveit![ros topics]," *IEEE Robotics & Automation Magazine*, vol. 19, no. 1, pp. 18–19, 2012.

[42] R. Margolin, L. Zelnik-Manor, and A. Tal, "How to evaluate foreground maps," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 248–255.

[43] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours," in *IEEE International Conference on Robotics and Automation*, 2016, pp. 3406–3413.

[44] J. Watson, J. Hughes, and F. Iida, "Real-world, real-time robotic grasping with convolutional neural networks," in *Annual Conference Towards Autonomous Robotic Systems*, 2017, pp. 617–626.

[45] Q. Lu, K. Chenna, B. Sundaralingam, and T. Hermans, "Planning multi-fingered grasps as probabilistic inference in a learned deep network," in *International Symposium on Robotics Research*, 2017.

[46] J. Mahler, M. Matl, V. Satish, M. Danielczuk, B. DeRose, S. McKinley, and K. Goldberg, "Learning ambidextrous robot grasping policies," *Science Robotics*, vol. 4, no. 26, pp. 49–84, 2019.

[47] M. Gualtieri, A. Ten Pas, K. Saenko, and R. Platt, “High precision grasp pose detection in dense clutter,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016, pp. 598–605.

[48] P. Ni, W. Zhang, W. Bai, M. Lin, and Q. Cao, “A new approach based on two-stream cnns for novel objects grasping in clutter,” *Journal of Intelligent & Robotic Systems*, vol. 94, no. 1, pp. 161–177, 2019.