Dex-Net MM: Deep Grasping for Surface Decluttering with a Low-Precision Mobile Manipulator

Benno Staub[1,2], Ajay Kumar Tanwani[1,3], Jeffrey Mahler[1,3], Michel Breyer[2], Michael Laskey[4], Yutaka Takaoka[4], Max Bajracharya[4], Roland Siegwart[2], Ken Goldberg[1,3,5]

***Abstract*— Surface decluttering in homes and machine shops can be performed with a mobile manipulator that recognizes and grasps objects in the environment to place them into corresponding bins. In contrast to fixed industrial manipulators, mobile robots have low-precision sensors and actuators. In this paper, we modify the Dex-Net 4.0 grasp planner to adapt to the parameters of the mobile manipulator. Experiments on grasping objects with varying shape complexity suggest that the resulting policy, Dex-Net MM, significantly outperforms both Dex-Net 4.0 and a baseline that aligns the grasp axis orthogonally to the principal axis of the object. In a surface decluttering experiment where the objects are randomly selected from 40 common machine shop objects, the robot is able to recognize, grasp and place them into the appropriate class bins 117 out of 135 trials (86.67% including 15 detected grasp failures and recovery on retry).**

I. Introduction

Mobile manipulation robots such as the Toyota Human Support Robot (HSR) [1] and the Fetch Robot [2] are emerging for human service applications such as decluttering the floor of a machine shop. The utility of mobile manipulators in surface decluttering depends upon the ability of the robot to reliably recognize and grasp various novel objects.

Dex-Net 4.0 [3] is a state-of-the-art grasp planner that plans robust grasps for a large variety of objects. The method combines simulation of thousands of 3D object models, analytical wrench mechanics, structured domain randomization and synthetic point clouds to train a deep learning optimization system. The learned policy rapidly processes high-resolution depth images to compute robust robot pick points on a diverse set of objects for an industrial manipulator on a fixed base.

However, due to inherent cost and weight limits, mobile manipulators have far lower precision in sensing and control than a fixed-based robot system. This makes reliable grasping challenging. The Toyota HSR [1] used in this paper has an Xtion Pro Live RGB-D camera with a 480×640 image resolution mounted on a mobile base. The camera must view objects from angles of 14° or more from the vertical due to the position of the camera with respect to the HSR base, leading to depth noise of $[2, 10]$mm depending on the pixel location in the depth image. Furthermore, the Toyota HSR has a 4DOF arm, a 1DOF torso lift joint and a 3DOF mobile base with a parallel-jaw gripper of $135mm$ maximum throw.

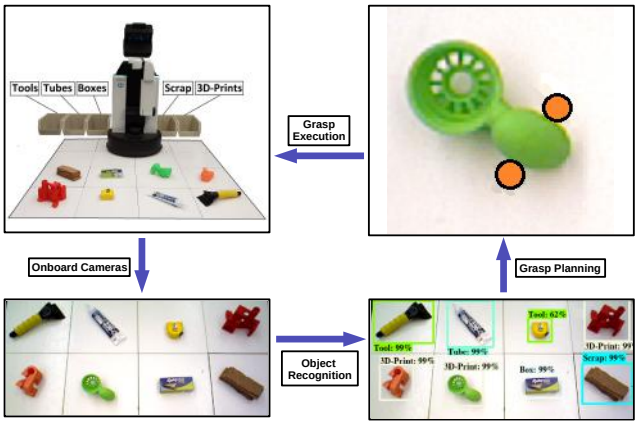

Fig. 1: System overview: Top left: Toyota HSR robot with 8 sample objects and 5 bins. Bottom left: The robot perceives the scene from the Xtion Pro Live RGB-D sensor. Bottom right: Domain Invariant Object Recognition (DIOR) model performs object recognition. Top right: An object is chosen and the grasp planner plans a grasp. The robot executes the grasp with DexNet-MM and places the object into the associated bin.

This work extends upon previous work on surface decluttering by Tanwani et al. in [4], where deep domain invariant models for object recognition and grasp planning are deployed on mobile robots for surface decluttering. This paper makes two contributions:

1) Dex-Net MM, a grasp planner developed for a mobile manipulator with low precision in perception and control.
2) Data from experiments on surface decluttering of 40 common machine shop objects in 5 classes with a Toyota HSR robot.

II. Related Work

A. Mobile Grasp Planning

On fixed-based manipulators, data-driven deep neural networks trained on large amounts of empirical data labelled with both physical robots [5] and simulation [6] lead to significant performance improvements [7], [8], [9]. One challenge with mobile manipulators is higher uncertainty in the end-effector pose due to base movement. A standard way to mitigate this imprecision is to place the mobile

[1]The AUTOLAB (automation.berkeley.edu), UC Berkeley, Berkeley, CA 94720, USA; Email:{ajay.tanwani, jmahler, goldberg}@berkeley.edu

[2] Autonomous Systems Lab, ETH Zurich, Zurich, CH-8092, Switzerland; Email: benno.staub@alumni.ethz.ch, michel.breyer@mavt.ethz.ch, rsiegwart@ethz.ch

[3]Department of Electrical Engineering and Computer Sciences, UC Berkeley, Berkeley, CA 94720, USA

[4]Toyota Research Institute, Los Altos, CA 94022, USA; Email: {michael.laskey, yutaka.takaoka, max.bajracharya}@tri.global

[5]Department of Industrial Engineering and Operations Research, UC Berkeley, Berkeley, CA 94720, USA

Fig. 2: The 40 objects commonly found in machine shops that were used for experiments. Bin categories: Tools, Tubes, Boxes, Scrap and 3D-Prints.

base in such a way that the manipulability of the arm is maximized and the grasp can be executed with limited base motion [10], [11]. This approach is impractical for tasks that require significant movement such as surface decluttering [4], bed-making [12] or biotechnological applications [13]. Thus, robust grasp planning policies are needed.

Different versions of the PCA Grasp Planner that grasps near the centroid of the object and orthogonal to its principal axis are common heuristics for such tasks [4], [14] because a grasp near the object centroid is often robust with respect to motion imprecision. More sophisticated grasp planners introduce probability distributions over object and robot pose to consider noisy sensing and execution. Applying bayesian filtering to such pose distributions can be used to continuously adapt the planned trajectory with model predictive control [15]. Action-Related Places (ARPlaces) — a collection of robot pose locations each with an assigned probability of success for a given action — are introduced in [16]. By successively updating the ARPlaces and incorporating them in the grasp planning policy, higher grasp reliability on a mobile manipulator has been achieved. Reinforcement learning is another powerful tool to incrementally adapt the control input to the current state and account for errors due to noise [17], [18].

B. Low-Precision Sensing

Low-cost mobile manipulators that do not rely on high-precision sensing for reliable object grasping have potential to assist people with limited motor abilities [12], [19], [20]. Gupta et al. [21] built such low-cost robots and put them into homes to collect real-world data for grasping. The resulting training dataset consisted of noisy and mis-labelled sensor readings because of commercial sensors and uncontrolled lightning conditions during collection. A grasp planner trained on this data outperformed Dex-Net 2.0 [6] in home environments.

To account for the higher imprecision, the grasp planning policy in [21] models sensor noise as a latent variable which can be marginalized out to plan more robust grasps. Gaussian Processes are another way to deal with imprecise sensor readings [22]. Uncertainty about the object contours is incorporated in the covariance matrix which decreases via iterative regrasping. Other approaches align the sensor input with known objects [23], shape primitives [24] or bounding boxes [25] with precomputed reliable grasps instead of planning on the noisy object shape given by the sensor reading. This makes it difficult to plan a grasp on objects for which the alignment fails.

Overcoming this issue, Morrison et al. [26] use a small and efficient neural network to predict grasps at each pixel of the image independent of both the object contour and pose. Johns et al. [27] smooth their discretized grasp function with Gaussian noise. The resulting grasps are robust with respect to the uncertainty of their system due to noisy joint encoders, camera miscalibration and kinematic deformation of links.

This work builds on previous work in [4], in which deep models for object recognition and grasp planning are learned from synthetic images in simulation, adapted to the real images of the robot environment by learning domain-invariant feature representations, and subsequently deployed for low-latency serving with Fog Robotics. Learning deep domain-invariant models by sim-to-real transfer reduces the need of collecting massive training data from the robot, while deploying the models on nearby resources enables prediction serving at less than 100 milliseconds. This paper focuses on the development of the grasp planning policy for mobile manipulators with low precision perception and control which was used in [4].

III. Problem Statement

We consider the problem of surface decluttering: using a mobile robot with a parallel-jaw gripper to iteratively grasp a single object from a planar worksurface and place the object in a receptacle based on its semantic category (e.g., tool, scrap). We assume that the mobile manipulator plans grasps based on images from a noisy RGB-D sensor.

Consider a robot decluttering m objects $o_1, ..., o_m$, each of which belongs to one of k categories $c_1, ..., c_k$ (e.g., tools, bottles, scrap). Each category corresponds to a unique receptacle. The goal is to transport each object into the receptacle corresponding to its category.

Given a state $\mathbf{x}_t$ at time t consisting of the geometry and pose of the objects, the robot receives a noisy observation $\mathbf{y}_t$ from its sensors. Based on this observation, the robot uses a policy π to plan an action $\mathbf{u}_t = \pi(\mathbf{y}_t)$ consisting of a 4DOF gripper pose (x, y, z, θ) and object category c_j. We model $\pi(\mathbf{y}_t)$ as the composition of a target object recognition policy and grasp planning policy:

$$\pi(\mathbf{y}_t) = \pi_{grasp} \circ \pi_{obj}(\mathbf{y}_t) = \pi_{grasp}(\pi_{obj}(\mathbf{y}_t))$$

The robot executes $\mathbf{u}_t$ by moving the gripper to the desired grasp pose (x, y, z, θ), closing the jaws, lifting the object, and transporting it to the receptacle corresponging to category c_j. The robot receives a binary reward $R(\mathbf{x}_t, \mathbf{u}_t) = 1$ if the robot places a single object into the correct receptacle at time t and $R(\mathbf{x}_t, \mathbf{u}_t) = 0$ otherwise.

The objective of this paper is to find a policy π to maximize the number of objects put in the correct receptacle:

$$\max \sum_t R(\mathbf{x}_t, \pi(\mathbf{y}_t))$$

We develop a policy that identifies objects using an object detection network trained with sim-to-real transfer learning and plans grasps using a Grasp Quality Convolutional Neural Network trained using Dex-Net with stochastic models of low-precision sensing and control.

IV. System Architecture

The Toyota HSR [1] shown in Fig. 1, is a compact mobile manipulator designed for human service applications. This robot is equipped with a commercially available Xtion Pro Live RGB-D sensor in its rotary head, a small range 4DOF manipulator arm with an angular jaw gripper — meaning the jaws close along an arc, a 1DOF torso lift joint and an omnidirectional mobile base. In this section, we quantify the imprecision of this robot both in sensing and manipulation. We then use the resulting uncertainties to modify the robot execution policy to limit uncertainty and adapt Dex-Net 4.0 to the new setting yielding Dex-Net MM, a robust grasp planning policy for a mobile manipulator.

A. Low-Precision Sensing

At time t, the robot acquires an observation $\mathbf{y}_t$ consisting of an RGB image $\mathcal{I}_t^c \in \mathbb{R}^{480\times640\times3}$ and a noisy depth image $\mathcal{I}_t^d \in \mathbb{R}^{480\times640}$. We use the standard deviation σ of the noise in consecutive depth images from a static scene as performance metric to quantify the sensing precision of the robot. Then we adapt the robot configuration to perceive the scene in a way to minimize sensor noise for more robust grasp planning.

To quantify the noise of the Asus Xtion Pro Live depth sensor on the HSR, we placed 24 identical objects o_i uniformly across the whole sensor image and read 30 depth images $\mathcal{I}_t^d$ with a frequency of 0.5Hz. The standard deviation $\sigma_{k,l}$ for the pixel with coordinates (k, l) showed that noise increases with the pixel distance to the image center from $\sigma_{240,320} \approx 2$mm in the center up to ≈ 10mm near the image corners. This confirms a more detailed sensor noise study from Rauscher et al. [28] that additionally quantified a growth of σ by 4mm per meter increased object to camera distance. The robot needs to perceive the scene close to the objects and ideally center the image around the target object for accurate sensing. The only way to achieve this in our scenario is a steep camera view that limits the field of view. A camera elevation angle of $14°$ — where $0°$ means an overhead camera — is the steepest angle that the HSR can perceive its complete workspace as illustrated in Fig. 3. This results in an average object depth value of 0.95m in the depth image. In addition, the graspable zone as explained in Sect. IV-B is put into the vertical center of the image yielding an optimal configuration for grasp planning near this area.

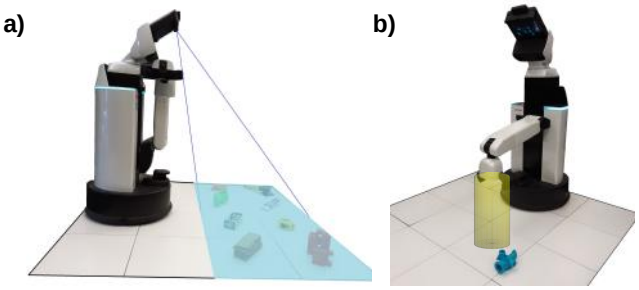

Fig. 3: a) The HSR RGB-D camera observes the floor from an angle $14°$ from the vertical. b) With a fixed base and allowing gripper height and rotation actuation only, a graspable cylinder for the end-effector can be projected in the scene. Aligning this cylinder with the target object minimizes additional base movement.

B. Low-Precision Movement

We quantify the uncertainty τ_{IK} of the HSR inverse kinematics (IK) controller [29] by giving a target 4DOF planar gripper pose on the grid in our workspace as input. After the robot executes the motion, we measure the deviation in gripper pose. The error in X and Y direction with the resulting Euclidean error for this controller is shown in the first row of Table I. These values are too high to achieve reliable parallel-jaw grasping. We tune this controller to reduce base movement and focus on manipulator arm movement by increasing the weight of base movement in the cost function that is used to plan end-effector trajectories. We similarly increase the weight of base rotation against base translation. This decreases the Euclidean error to ±7.3mm as can be seen in the second row of Table I. This is still too high for reliable parallel-jaw grasping. With a fixed base and moving only the wrist roll joint and translational arm height joint, the graspable area of the gripper can be formulated as a cylinder $V \in \{r, h\}$ with r the radius defined by the maximum gripper width and h the height as shown in Fig. 3. We calibrate a predefined manipulator movement allowing the robot to execute grasps perpendicular to the ground at any height and gripper rotation in the cylinder with minimal base movement. Base movement is used to align this cylinder

Controller	Error in X	Error in Y	Euclidean Error
τ_{IK}	6.9mm	7.9mm	10.5mm
$\tau_{IK,tuned}$	5.9mm	4.3mm	7.3mm
τ_{cyl}	**2.6mm**	**3.0mm**	**4.0mm**

TABLE I: End-effector motion imprecision for different control methods.

with the desired object. With the object in the graspable zone of the gripper, it is also in the center of the depth image. From this configuration, a grasp is planned and the robot executes it with minimal additional base movement. The error for the developed control algorithm is shown in the last row of Table I.

C. Target Object Recognition Policy

We leverage on the pretrained features of the Domain Invariant Object Recognition (DIOR) model [4] that are learned across 20K simulation images of cluttered piles sampled from 770 unique 3D object meshes and 212 real images of cluttered piles sampled from 102 physical objects, see [4] for further details. We use the DIOR_dann model that shares parameters of the feature representation for both the sim and the real domain and outputs the bounding box with corresponding class label c_j for each object recognized in the image. We use the MobileNet Single-Shot MultiBox Detector (SSD) [30], [31] algorithm with focal loss and feature pyramids for adapting the model to the new object categories. Training adapts the model parameters such that the classification loss $\mathcal{L}_{y_c}$ of predicting the correct object class c_j and the localization loss $\mathcal{L}_{y_l}$ of predicting the bounding box locations is minimized over all images.

We split 40 objects across 5 class categories $\{c_j\}_{j=1}^{5}$ including Tools, Tubes (glue, caulk,..), Boxes (for batteries, staplers, nails,..), 3D-Prints (representing assembly parts) and Scrap (cloth, sponge,..), see Fig. 2. We collect a dataset of 280 real RGB images $\{\mathcal{I}^c_{t,real}\}_{t=1}^{280}$ and hand-label bounding boxes and object categories for each image. With this dataset, we adapt the pre-trained DIOR_dann model to our objects.

Our target object recognition policy $\pi_{obj}(\mathbf{y}_t)$ feeds the RGB image $\mathcal{I}^c_t$ as input to the adapted DIOR_dann model and chooses the nearest recognized object as target. It uses the bounding box of the target object to crop the depth image $\mathcal{I}^d_{t,crop} = \pi_{obj}(\mathcal{I}^d_t)$ as input for the grasp planning policy.

D. Grasp Planning Policy

Given the cropped depth image, our grasp planning policy outputs a 4DOF gripper pose $\pi_{grasp}(\mathcal{I}^d_{t,crop}) = (x, y, z, \theta)$ consisting of the grasp center point and planar orientation of the gripper. We now present Dex-Net MM – an adaptation of Dex-Net 4.0 [3] grasp planning model to a mobile base with both low-precision sensing and manipulation.

1) Transfer Learning: We train the grasping model in the simulation environment. A virtual camera points towards a planar worksurface, objects are dropped randomly into heaps using dynamic simulation, and a synthetic depth image is rendered. Grasps are evaluated using a robust wrench resistance metric in simulation. Instead of evaluating the exact intended grasp (x, y, z, θ), we perturb the grasp position

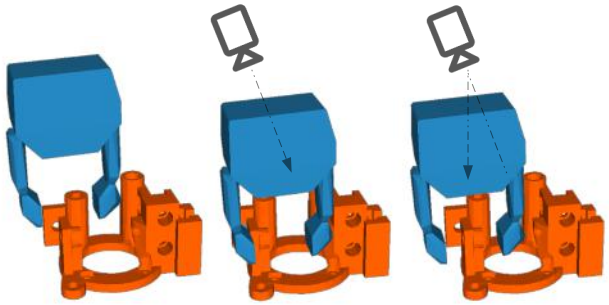

Fig. 4: Given an initial state (left), Dex-Net 4.0 adds a depth offset to the grasp to lower the gripper. This results in collision with the object because of the tilted depth vector as shown in the middle figure. Right: Dex-Net MM uses an additional height offset in the grasp parametrization to lower the grasp in the 3D space (right).

using our knowledge about the gripper imprecision. For each grasp, an error in x and y is sampled from a Gaussian distribution and the translated grasp $(x + e_x, y + e_y, z, \theta)$ is evaluated. The standard deviation τ_{DNMM} has been chosen to adapt to the measured values from Table I. We chose a value of $\tau_{DNMM} = 2.5$mm. The tested grasps are then projected onto the rendered depth image which results in a large set of labelled grasps. These labels are used as training data for the grasp planning model. In our setting, the HSR perceives its environment significantly different from the dataset generated with a vertical high-resolution camera which has been used to train Dex-Net 4.0. In addition, the distance to objects varies depending on the robot's pose in the workspace. To adapt the Dex-Net 4.0 grasp planning policy to this setting, we generated a new dataset with 1000 simulated scenes. For each scene, 3-10 out of 5000 unique 3D objects are dropped from a 15cm height into a wide bin and 10 different camera poses are sampled from which a synthetic depth image was rendered using the intrinsics of a Xtion Pro Live camera. Each camera pose is uniformly sampled from the set of configurations that point towards the object heap with a camera elevation angle in $[10, 15]^\circ$ and camera to heap distance in $[0.75, 1.25]$m. We then sampled and evaluated 260 grasps for this scene using an algorithmic supervisor. Each grasp is projected onto the 10 rendered depth images yielding 10 labelled datapoints per grasp and scene. This results in 2.6million $(1000 \times 10 \times 260)$ datapoints for our dataset. We finetuned the existing Dex-Net 4.0 model for 50 epochs on this dataset using a batch size of 64 images.

2) Tilt Offset Correction: The grasp sampler of Dex-Net models first samples antipodal pairs from edge points on the object surface. Using the center of these antipodal pairs for the grasp parametrization results in a gripper pose above the object, as shown in Fig. 4. From this reference pose, it lowers the gripper to a target pose from which the object can be grasped. In a vertical camera setting, depth and height can be used interchangeably. This assumption is exploited by Dex-Net 4.0 in the grasp sampling process where it adds a depth offset uniformly sampled from $[5, 40]$mm to lower the reference grasp. Changing the depth value in a tilted camera setting lowers the grasp in the z direction and moves the gripper away from the robot. This leads to gripper collisions

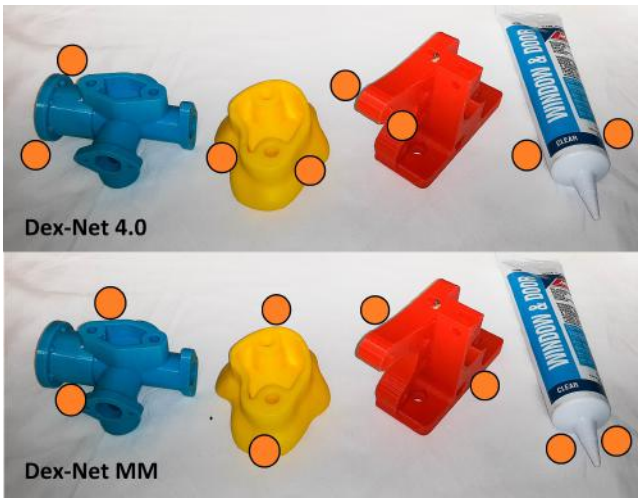

Fig. 5: Planned grasps for different methods. Top row: Due to gripper imprecision, the grasps planned by Dex-Net 4.0 are prone to fail. Bottom row: Dex-Net MM plans more conservative grasps in all cases except the last one in which it preferred a near grasp limiting base movement over a conservative grasp near the object center.

with the object as demonstrated in Fig. 4. Additionally, the planner becomes biased towards grasping too high because the distance the gripper is lowered is shrunk with respect to the depth offset.

We develop a grasp sampler to account for these limitations. Instead of reasoning in the camera reference frame, this grasp sampler acts in the 3D world frame and adds a height offset value uniformly sampled from $[15, 25]$mm to the grasp parametrization to lower the grasp from the reference pose as illustrated in Fig. 4. In addition, the gradient threshold for detecting edge pixels on the objects used for antipodal pair sampling is decreased from 5mm to 3mm because in a tilted camera setting, the front edge cannot be clearly seen as obvious depth jumps are missing. The lower threshold results in the detection of edge pixels in the depth image at noisy areas and therefore grasps are also sampled on the floor. This is not a problem because Dex-Net MM prunes out these grasps due to their low quality values.

V. Experiments and Results

We conducted physical experiments using the Toyota HSR. We use Dex-Net 4.0 [3] and the PCA Grasp Planner [4] as baselines to compare grasping reliability for 125 grasp attempts on 25 objects with varying surface complexity in the first experiment. In the second experiment, we test the surface decluttering performance of the complete pipeline with 12 trials on putting 40 common machine shop objects into bins.

A. Grasping

We created a dataset of 25 objects with various surface complexity (10 of Level 1, 8 of Level 2, 7 of Level 3) [3]. Level 1 objects have basic geometric shapes, such as boxes, bottles and spheres while Level 2 objects have more complex contours such as the adversarial objects used in [6]. Objects that are empirically very hard to grasp are chosen as Level 3 objects.

To measure grasp performance, we placed a single object in front of the robot similar to Fig. 3. With the RGB-D sensor directed towards the object, we planned a grasp for the robot to pick the object and used the developed controller from Sect. IV-B to execute it. If the robot was able to grasp the object, lift it 30cm, and hold it for 5 seconds, the trial was counted as success. Fig. 6 shows the success rates for each method separated by object complexity level. The PCA Grasp Planner achieved 87.5% reliability on the Level 1 objects. Its performance drops down to 77.5% and 48.6% for Level 2 and Level 3 objects respectively. The centroid of the object may not be a good choice for irregular shapes. Dex-Net 4.0 was able to plan reliable grasps in the image space but often grasped too high or collided with the object which resulted in a 54% success rate on Level 1 objects. Its performance increased to 85% for Level 2 objects. The size of Level 2 objects may be the cause for this, as bigger objects seem to be less sensitive to height errors. On Level 3 objects, the success rate of Dex-Net 4.0 drops to 68.6%. Dex-Net MM outperforms both baselines in this setting with no failed grasps for Level 1 and 2 objects and the policy succeeded on 85.7% of grasps for Level 3 objects.

B. Surface Decluttering

In this experiment we evaluate the performance of the surface decluttering pipeline. The HSR is situated in a small rectangular workspace of $120cm \times 120cm$. Objects are placed in front of the robot and the corresponding bins are behind the robot. An overview of the setup can be seen on the left of Fig. 7. Addressing this task in a machine workshop, we created a dataset of 40 common machine shop objects shown in Fig. 2. The following 5 classes are represented by 8 objects each: Tool, Tube, Box, 3D-Print and Scrap. We ran 12 trials. For each trial we chose 10 objects at random (2 per class) and placed them in the workspace of the robot. For this experiment, each object was singulated. Two input images from example scenes can be seen on the right of Fig. 7. Given the RGB-D input from the camera, the robot's task was to classify the objects in the scene and choose an object. It then executed a grasp planned by Dex-Net MM on the target object and put it into the corresponding bin. Using the binary reward $R(\mathbf{x}_t, \mathbf{u}_t)$ from Sect. III, the system achieved a score of $117/120$ objects placed in the correct receptacle after 135 grasp attempts. We evaluated the performance of the grasping and object classification for each class separately. For grasping success, we counted the number of successful grasps and divided this by the number of times the robot attempted to grasp an object from this particular class over the 12 trials. For classification, we checked the number of true positives and false positives in the corresponding bin at the end of each trial and summed them up, with a maximum of 24 true positives and 0 false positives per class. Table II shows the results. With an overall grasping success rate of 88.9% (120 successful grasps out of 135 attempts), the Dex-Net MM policy was able to successfully declutter all objects

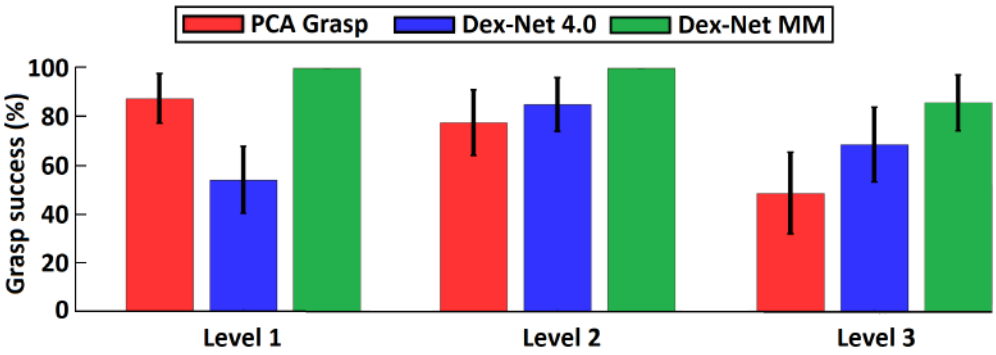

Fig. 6: Results from 5 grasp attempts on 10 Level 1, 8 Level 2, and 7 Level 3 objects for each policy on objects with different shape complexity.

Fig. 7: Left: Top view from setup for surface decluttering experiment. Right: Two sample input images taken from the experiment.

Fig. 8: Three example failure modes from Dex-Net MM. Left: The object was lifted successfully, but the screwdriver's weight caused it to rotate and then it slipped out of the gripper during transportation. Center: The robot grasped slightly more below than planned and the object slipped because of the round surface. Right: Due to motion imprecision, the HSR grasped further below than planned and failed to grasp the object. Such counter examples may be due to limited positive grasps.

using 15 retries. For boxes, all grasps succeeded. 3D-Prints were the hardest objects for the HSR to grasp with 77.4% success rate. Three example failure modes of the developed system are shown in Fig. 8. With 7 of the 15 failed grasps, error in z (height) was the main failure mode. The other failures were due to poor grasp planning (4), imprecise motion (3) and ungraspable objects (1). The last 3 failed grasps achieved lifting the object, but the grasp was unstable and the object dropped during its way to the bin. With isolated objects, the object classification model misclassified 3 objects out of 120. A screwdriver was misclassified twice as a tube and the yellow measurement tape was classified as a 3D-print. Fig. 9 depicts the general pattern in highlighting the differences in the Dex-Net MM. Increasing the motion

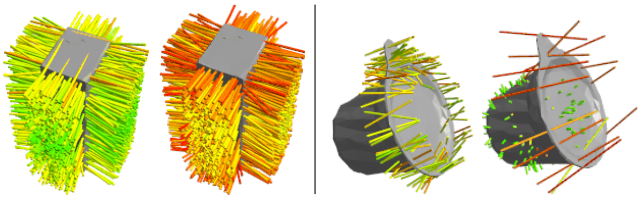

Fig. 9: Grasp axes colored by predicted robustness (green is high, red is low):*(left)* Effect of increasing motion imprecision on grasp labels in the training dataset on reducing grasp quality, *(right)* Effect of increasing gripper throw on grasp labels in the training dataset. The colors in far right are rescaled showing that the grasps around the base are more robust than those at the rim.

imprecision encourages the grasps to be focused towards the center of the object (more robust); while increasing grasp width allows for more potential grasps to be available.

Class	Grasping			TP	FP
Tool	96.0%	±	7.7%	21	0
Tube	92.3%	±	10.2%	24	2
Box	100.0%			24	0
3D-Print	77.4%	±	14.7%	24	1
Scrap	92.3%	±	10.2%	24	0

TABLE II: Experimental results for 120 surface decluttering trials with 24 objects for each class. Left column: Grasping success rate with standard error of the mean. Right column: Number of true and false positives for objects put into the bins. Less than 24 true positives means that some objects from this class were put into another bin. More than 0 false positives means that an object from another class was put into this bin.

C. Timing Analysis

Surface decluttering is not a time critical task. The average time needed by our developed pipeline to grasp an object and put it into its respective bin in the setup explained in Sect. V-B is 76.1sec. This results in 47 objects per hour. The average duration for each subprocess is shown in Table III.

Policy Stage	Runtime
Image Segmentation	4.2s
Grasp planning	2.6s
Grasp execution	19.7s
Drop into bin	22.9s
Go back to start	26.7s
Total	**76.1s**

TABLE III: Average time for each subprocess in surface decluttering experiment.

VI. Conclusions

We present Dex-Net MM, a grasp planning policy for a mobile manipulator with low precision perception and

control, and use it with a deep domain invariant object recognition model for surface decluttering. In physical experiments with 40 objects commonly used in homes and machine shops, the pipeline was able to successfully declutter singulated objects with 88.9% overall grasping success rate of $117/120$ objects put into the correct bin.

VII. Acknowledgments

This research was performed at the AUTOLAB at UC Berkeley in affiliation with the Berkeley AI Research (BAIR) Lab, Berkeley Deep Drive (BDD), the Real-Time Intelligent Secure Execution (RISE) Lab, the CITRIS "People and Robots" (CPAR) Initiative, and the Scalable Collaborative Human-Robot Learning (SCHooL) Project, NSF National Robotics Initiative Award 1734633 and by the NSF ECDI Secure Fog Robotics Project Award 1838833. The authors were supported in part by donations from Siemens, Google, Amazon Robotics, Toyota Research Institute, Autodesk, ABB, Knapp, Loccioni, Honda, Intel, Comcast, Cisco, Hewlett-Packard and by equipment grants from PhotoNeo, NVidia, and Intuitive Surgical. We thank our colleagues for feedback, in particular Aditya Ganapathi, Roy Fox, Daniel Seita, Mike Danielczuk, Matt Matl, Jackson Chui, Kate Sanders, Sean Huang, Jessica Ji, Ryan Hoque, and William Wong.

References

[1] T. Yamamoto, K. Terada, A. Ochiai, F. Saito, Y. Asahara, and K. Murase, "Development of the research platform of a domestic mobile manipulator utilized for international competition and field test," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Oct 2018.

[2] M. Wise, M. Ferguson, D. King, E. Diehr, and D. Dymesich, "Fetch freight : Standard platforms for service robot applications," in *IJCAI Workshop on Autonomous Mobile Service Robot*, 2016.

[3] J. Mahler, M. Matl, V. Satish, M. Danielczuk, B. DeRose, S. McKinley, and K. Goldberg, "Learning ambidextrous robot grasping policies," *Science Robotics*, Jan 2019.

[4] A. K. Tanwani, N. Mor, J. Kubiatowicz, J. E. Gonzalez, and K. Goldberg, "A fog robotics approach to deep robot learning: Application to object recognition and grasp planning in surface decluttering," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2019.

[5] S. Levine, P. Pastor, A. Krizhevsky, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *Int. Journal of Robotics Research (IJRR)*, 2017.

[6] 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," *Proc. Robotics: Science and Systems (RSS)*, 2017.

[7] A. Saxena, J. Driemeyer, J. Kearns, and A. Y. Ng, "Robotic grasping of novel objects," in *Proc. Advances in Neural Information Processing Systems*, 2007.

[8] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige, S. Levine, and V. Vanhoucke, "Using simulation and domain adaptation to improve efficiency of deep robotic grasping," *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2018.

[9] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *Int. Journal of Robotics Research (IJRR)*.

[12] D. Seita, N. Jamali, M. Laskey, R. Berenstein, A. K. Tanwani, P. Baskaran, S. Iba, J. Canny, and K. Goldberg, "Robot Bed-Making: Deep Transfer Learning Using Depth Sensing of Deformable Fabric," *CoRR*, 2018.

[10] F. Chen, M. Selvaggio, and D. G. Caldwell, "Dexterous grasping by manipulability selection for mobile manipulator with visual guidance," *IEEE Transactions on Industrial Informatics*, Feb 2019.

[11] S. Wang, H. Guo, Y. Huang, A. Ye, and K. Yuan, "Path planning for nonholonomic mobile manipulators grasping based on multi-objective constraint," in *36th Chinese Control Conference (CCC)*, July 2017.

[13] J. Zhang, O. Onaizah, K. Middleton, L. You, and E. Diller, "Reliable grasping of three-dimensional untethered mobile magnetic microgripper for autonomous pick-and-place," *IEEE Robotics & Automation Letters*, April 2017.

[14] M. Logothetis, G. C. Karras, S. Heshmati-Alamdari, P. Vlantis, and K. J. Kyriakopoulos, "A model predictive control approach for vision-based object grasping via mobile manipulator," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Oct 2018.

[15] D. Chen and G. von Wichert, "An uncertainty-aware precision grasping process for objects with unknown dimensions," *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2015.

[16] F. Stulp, A. Fedrizzi, L. Mösenlechner, and M. Beetz, "Learning and reasoning with action-related places for robust mobile manipulation," *Journal of Artificial Intelligence Research*, 2012.

[17] Z. Li, T. Zhao, F. Chen, Y. Hu, C. Su, and T. Fukuda, "Reinforcement learning of manipulation and grasping using dynamical movement primitives for a humanoidlike mobile manipulator," *IEEE/ASME Transactions on Mechatronics*, Feb 2018.

[18] Y. Wang, H. Lang, and C. W. de Silva, "A hybrid visual servo controller for robust grasping by wheeled mobile robots," *IEEE/ASME Transactions on Mechatronics*, Oct 2010.

[19] M. Gualtieri, J. Kuczynski, A. M. Shultz, A. ten Pas, R. P. Jr., and H. A. Yanco, "Open-world assistive grasping using laser selection," *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2017.

[20] A. Jain and C. C. Kemp, "El-e: an assistive mobile manipulator that autonomously fetches objects from flat surfaces," *Autonomous Robots*, Sep 2009.

[21] A. Gupta, A. Murali, D. Gandhi, and L. Pinto, "Robot learning in homes: Improving generalization and reducing dataset bias," in *Proc. Advances in Neural Information Processing Systems*, 2018.

[22] S. Dragiev, M. Toussaint, and M. Gienger, "Uncertainty aware grasping and tactile exploration," *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2013.

[23] S. Srinivasa, D. Ferguson, J. M. Vandeweghe, R. Diankov, D. Berenson, C. Helfrich, and K. Strasdat, "The robotic busboy: Steps towards developing a mobile robotic home assistant," in *International Conference on Intelligent Autonomous Systems*, July 2008.

[24] M. Nieuwenhuisen, D. Droeschel, D. Holz, J. Stckler, A. Berner, J. Li, R. Klein, and S. Behnke, "Mobile bin picking with an anthropomorphic service robot," *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2013.

[25] J. Stückler, R. Steffens, D. Holz, and S. Behnke, "Efficient 3d object perception and grasp planning for mobile manipulation in domestic environments," *Robot. Auton. Syst.*, Oct. 2013.

[26] D. Morrison, P. Corke, and J. Leitner, "Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach," *Proc. Robotics: Science and Systems (RSS)*, 2018.

[27] E. Johns, S. Leutenegger, and A. J. Davison, "Deep learning a grasp function for grasping under gripper pose uncertainty," *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2016.

[28] G. Rauscher, D. Dubé, and A. Zell, "A comparison of 3d sensors for wheeled mobile robots," in *Intelligent Autonomous Systems*, 2014.

[29] K. M. Lynch and F. C. Park, *Modern Robotics: Mechanics, Planning, and Control*. New York, NY, USA: Cambridge University Press, 1st ed., 2017.

[30] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision – ECCV 2016*, 2016.

[31] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.