Learning Hardware Agnostic Grasps for a Universal Jamming Gripper

Yun Jiang, John R. Amend, Jr., Hod Lipson and Ashutosh Saxena

***Abstract*— Grasping has been studied from various perspectives including planning, control, and learning. In this paper, we take a learning approach to predict successful grasps for a universal jamming gripper. A jamming gripper is comprised of a flexible membrane filled with granular material, and it can quickly harden or soften to grip objects of varying shape by modulating the air pressure within the membrane. Although this gripper is easy to control, developing a physical model of its gripping mechanism is difficult because it undergoes significant deformation during use. Thus, many grasping approaches based on physical models (such as based on form- and force-closure) would be challenging to apply to a jamming gripper. Here we instead use a supervised learning algorithm and design both visual and shape features for capturing the properties of good grasps. We show that given target object data from an RGBD sensor, our algorithm can predict successful grasps for the jamming gripper without requiring a physical model. It can therefore be applied to both a parallel plate gripper and a jamming gripper without modification. We demonstrate that our learning algorithm enables both grippers to pick up a wide variety of objects, including objects from outside the training set. Through robotic experiments we are then able to define the type of objects each gripper is best suited for handling.**

Fig. 1: The universal jamming gripper shown choosing a successful grasp point on a target object using our hardware agnostic learning algorithm.

I. INTRODUCTION

There are several approaches that have been successfully implemented to solve the problem of robotic grasping. If the kinematics of the gripper are known and a 3D model of the object is available, we can use methods that consider form and force closure [1]–[3] to plan a grasp (e.g., GraspIt [4]). Closed loop feedback has also been used to perform grasps, both with visual feedback and tactile feedback [5], [6]. Most of these studies however assume a (often very detailed) physical model of the gripper.

Consider the universal jamming gripper [7], [8] shown in Fig. 1. The design and control of this gripper are very simple – it is comprised of a flexible outer membrane filled with granular material, and modulating the air pressure within the membrane hardens or softens the gripper to enable the gripping function. This gripper has proved capable at picking up a wide variety of objects in open loop experiments directed by a human operator, however, autonomous grasping using one of the previously mentioned methods would require that we develop a physical model of its gripping behavior. Specifying such a model would be very difficult because of the deformation the jamming gripper undergoes when contacting a target object.

In this paper we present a solution to this problem by applying a hardware agnostic grasp learning algorithm to a jamming gripper. We call our algorithm hardware agnostic because it does not require or assume any physical model of the gripper. Our algorithm is motivated by recent work in learning techniques [9]–[13], in which learning algorithms are trained on a large amount of labeled data in order to generate robust grasping hypotheses, even on objects not included in the training set. However in this previous work, grasps are represented by one or a pair of grasping points and thus only applicable for two or three-fingered grippers.

Our approach instead uses a 'grasping rectangle' for representing jamming grasps. The rectangle can not only encode the physical space occupied by the fingers (in the case of traditional fingered grippers), but can also encode the contact area of the universal jamming gripper. Our algorithm first learns a ranking function that maps a grasping rectangle (represented by its feature vector) to a score of the likelihood of a successful grasp (using an SVM ranking algorithm [14]). Then, the trained algorithm is able to predict the highest-score grasp from a new 2D image and its aligned point cloud. To capture the distinction between proper and invalid grasps, we design 2D features from the RGB image and 3D shape features from the point cloud. Filters and fuzzy histograms are used to extract visual cues from the color image, while the normals and curvature at pixel level along with the Fast Point Feature Histogram [15] are extracted from the point

Yun Jiang and Ashutosh Saxena are with the Department of Computer Science, Cornell University, Ithaca NY 14853 USA {yunjiang, asaxena}@cs.cornell.edu

John R. Amend, Jr. is with the Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY 14853 USA jra224@cornell.edu

Hod Lipson is with the Sibley School of Mechanical and Aerospace Engineering and the Faculty of Computing and Information Science, Cornell University, Ithaca NY 14853 USA hod.lipson@cornell.edu

cloud.

We demonstrate through robotic experiments that we are able to learn successful autonomous grasps for both a jamming gripper and a parallel plate gripper without changing the learning framework. Our algorithm also outperforms a baseline heuristic method that always attempts to grip objects at their center. For some objects, stable grasps may vary between the two grippers due to their distinct gripping mechanisms, however, our algorithm can predict correct but different grasps for both grippers. Since our algorithm aims at learning hardware agnostic grasps, meaning no physical model of the gripper is required to grip objects, it can potentially be applied to many other kinds of grippers, and it can also help us compare grippers based on the types of objects each is best suited for handling.

II. RELATED WORK

Grasping with object geometry known. Much of the prior work in robotic grasping assumes complete knowledge of the target object geometry as well as the gripper geometry. From this geometric information, control and planning algorithms can be designed for successful grasping with force closure [1], [2] and form closure [3]. Several survey papers covering this type of work are available [16]–[18]. Niparnan and Sudsang [19] relaxed the assumption of access to full object geometry, and instead their algorithm searches for force-closure grasps on sampled points on the object's surface. Huebner et al. [20] transformed 3D models to box-based approximation before generating grasp hypotheses. Glover et al. [21] built generative probabilistic models for known objects when they are occluded or deformed to complete the object geometry. Some other work has focused on learning grasps from examples, such as [22], [23]. But they are limited to known objects.

Grasping with object geometry unknown. For grasping in real-world situations or unknown environments, complete geometric information is often unavailable. Others have addressed this by representing the object as a simpler known shape, or as a composition of known shapes with pre-computed grasp primitives [24]. Additional work has been done using object edges and contours to compute form and force closure [25]–[27].

Learning algorithms can generalize grasping models from a collection of objects, enabling successful grasps of previously unseen objects. Saxena et al. [9], [11] first proposed a image-based learning algorithm to predict a single grasping point, and with the help of other learning techniques [28] gripper orientation can also be estimated. Depth information, such as point cloud, has also been included to obtain higher performance [29]. Le et al. [12] suggested a more suitable representation for two-jawed grippers – a pair of grasp points. Rao et al. [30] utilized a segmented point cloud to enable grasping in cluttered environments. In fact, learning algorithms have also been successfully applied to other object handling tasks such as placing an object in an unstructured environment [31], [32] and opening doors by turning door handles [33]. These learning approaches show the possibility to handle the object without knowing the object geometry. However, it is unclear how they would perform when applied to a significantly different gripper.

Grasping with compliant grippers. It is a common practice in robot gripping to add some soft material to the gripping surfaces in order to achieve increased conformation to the target object [34], [35]. Simpson [36] presented a design that used pockets of granular materials for this purpose, and Schmidt [37] and Perovskii [38] each proposed designs where similar pockets of granular materials could also be vacuum hardened after deforming to produce a custom-contour gripping surface. Reinmueller and Weissmantel [39] worked on a design with similar vacuum-hardening pockets and suggested that a gripper with a single pocket of granules might be able to grip objects on its own. Recently this idea was explored in more detail [7], and a single pocket jamming gripper was presented. Further developments including the addition of positive pressure for improving performance and ejection of objects were recently presented [8]. The gripper we use here is based on this most recent design. Prior to this work however, jamming grippers have only been demonstrated with open loop control. This paper represents the first time a jamming gripper has been controlled with a vision based grasping algorithm.

III. GRASPING WITH JAMMING GRIPPERS

Our group at Cornell (Brown et al. [7] and Amend et al. [8]) has recently presented a design for a universal robot gripper called a jamming gripper. A jamming gripper is comprised of a flexible outer membrane filled with granular material. By exploiting the jamming transition of granular materials ([40]–[45]) through modulation of the air pressure within the membrane, our gripper can transform from a fluid-like state to a solid-like state. In the fluid state, our gripper passively deforms around a target object. The gripper then vacuum hardens to achieve the solid state and rigidly grip the object. Using this gripping mechanism, jamming grippers have had success gripping objects of widely varying shape, hardness, fragility, and texture, including multiple objects gripped at once [7], [8]. This work was also done in open loop control where the gripper location was given by a human operator.

Although the design of a jamming gripper is very simple, developing a model of its gripping behavior is especially difficult. Predicting how the gripper will contact and conform to a target object would require analysis of the object geometry and predictive models for the flow of the grains and the deformation of the membrane. Some insight to the granular deformation could perhaps be gathered from work in the areas of soil mechanics and especially critical state soil mechanics [46], but linking this with the elastic deformation of the membrane would likely require a physics engine simulation or finite element approach. Such a complex model would have limited utility for online grasp planning. Fortunately, jamming grippers have been shown to perform well without any such model. In open loop experiments—where the gripper is only given a location to perform the grasp action—jamming grippers have shown high reliability and error tolerance for gripping a wide range of objects [8]. This indicates that if we are able to design an algorithm that can predict well the location on the object to grasp, then our jamming gripper would be able to perform autonomous grasps.

An intuitive first approach is for the jamming gripper to always grasp at the center of the object. For small objects (smaller than the gripper itself), the error tolerance of the jamming gripper makes almost any location on the object a suitable grasp point. However for objects that are larger than the gripper in some dimension (for example a length of pipe), large torques or off-axis forces on a gripped object can lead to failure [7], so it is typically preferred that the center of mass of the object be located in line with the gripper's central axis. Problems with this strategy arise when the center of mass is not located within the object itself (for example in an L-shape), or when the center of mass is otherwise difficult to grip. There is no simple rule for weighing the tradeoff between minimizing torques and choosing a feature to grip, which motivates the use of a learning approach.

A second possible approach could be via planning or control based algorithms. These methods rely on the gripper's physical model and have been widely applied to multi-fingered grippers. Although they can generate accurate grasps given a specific gripper and complete 3D data, it is formidable to apply such an algorithm to a jamming gripper because of its malleability.

In this paper we consider only vertical grips with the jamming gripper (where the gripper approaches the object lying on a surface from above) because horizontal grips are only possible when a backstop is available to push against, or in circumstances where the object is heavy enough to not slide.[1]

IV. HARDWARE AGNOSTIC LEARNING ALGORITHM

In order to address the aforementioned problems, we propose a hardware agnostic learning algorithm. In this paper we call a method hardware agnostic if it does not require or assume any physical model of the gripper. For the jamming gripper, this kind of learning algorithm has two merits: 1) it bypasses possibly complicated models of gripper deformation; and 2) it can generalize a comprehensive model from a number of meaningful features and sufficient training data. The abilities of different grippers will thus be captured through relevant features rather than a physical model. Hence, we propose a hardware agnostic learning algorithm to facilitate the jamming gripper in grasping.

[1]In detail, if we consider a solid cube target object approximately half the size of the jamming gripper, the gripper would need to apply a contact force ≈25 N to the object as it deforms to its shape [8]. Even if we assume a coefficient of friction of 1, this cube would need a density of about 40,000 kg/m^3 to resist the contact force without sliding (five times the density of steel).

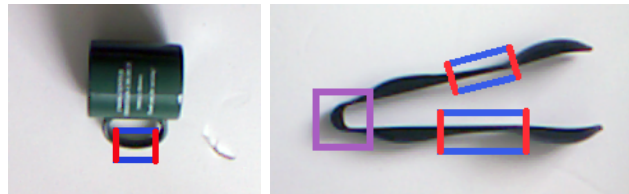

Fig. 2: Examples of grasping rectangles for different objects. The purple rectangle is only valid for the jamming gripper.

A. Grasp Representation

In the task of grasping, the goal is to find an optimal gripper configuration at the final grasping stage – when the gripper contacts the object. Our representation for a grasp is motivated by previous work [13], where a rectangle gives the location and orientation of gripper fingers in the image plane. In the case of a jamming gripper, we aim to find an oriented rectangle where the dimension of the rectangle approximates the area of contact rather than a finger location. Since all relevant features bounded by a rectangle are extracted to depict the corresponding grasp, grasping clues are more likely to be captured with this method. The size of the grasping rectangle is inferred by the learning algorithm, and therefore can change automatically to adapt to different sizes of jamming grippers. The most important benefit is that the rectangle representation needs no physical model from the gripper.

B. Learning Algorithm

Given an image and a point cloud, our algorithm aims to find the optimal grasping rectangle(s). To measure 'optimal' quantitatively, we construct a score function mapping any rectangle in the image (denoted by its feature vector) to a real number. Thus, our goal is to find the highest-score rectangle. Mathematically, for a rectangle G in the image I, $\phi(G) \in \mathbb{R}^k$ is defined as the feature vector of G of size k. Our score function is then defined to be a linear function of the features:

$$f(G) = w^T \phi(G) = \sum_{i=1}^{k} w_i \phi_i(G) \quad (1)$$

The parameter w is learned from manually-labeled data. We consider finding the optimal grasping rectangle as a ranking problem, and w is derived using an SVM ranking algorithm [14]. It is possible to find the highest scoring rectangle for an object by extracting the feature vector and calculating the score for each of the possible rectangles.

V. FEATURE EXTRACTION

The input to our algorithm is composed of an RGB image and a point cloud of distance values from a Microsoft Kinect sensor.The precision of the perceived point clouds is influenced by the texture, distance, occlusion and inherent sensor noise. Therefore we utilize features from both image intensity and the point cloud to obtain object geometry.

A. Image Features

In order to extract visual cues such as edges, texture, and color, we use features proposed by Saxena et al. [11]: nine Laws' masks and six oriented edge filters are applied to a transformed YCbCr image. Hence, each pixel has a feature vector of dimension 17. All filtered images are normalized from 0 to 1 to reduce illumination changes caused by different lighting conditions.

To capture properties for rectangles of all sizes, features need to be scale-invariant and capable of describing the distribution of the input. Histograms satisfy these criteria, so we employ normalized fuzzy histograms [47]. Compared with normal histograms, fuzzy histograms are more robust to small changes in input values. In a normal histogram, if an input value is near a bin boundary, a small change in the value can cause a shift from one bin to another. This means that values near boundaries are extremely sensitive to noise. To address this issue, fuzzy histograms are calculated based on (linear) fuzzy partitions. For each bin i, we define a bin center c_i, and we allocate each input value x to bins i, $i+1$ such that $c_i < x < c_{i+1}$ in the manner that bin i receives $1 - \frac{x-c_i}{c_{i+1}-c_i}$ and bin i+1 receives $\frac{x-c_i}{c_{i+1}-c_i}$. In this way, small changes in x would only cause a commensurate change in the fuzzy histogram. In total, 15 bins are equally spaced between 0 and 1 for each histogram. Every rectangle is also divided into 3 equal-sized horizontal strips [13]. This partitioning has the potential to recognize handle-like parts on objects, as the center strip looks different from the other two. In summary, we have a total of $3 \times 15 \times 17 = 765$ image features.

B. Point Cloud Features

From the point cloud, we calculate the normal vector and curvature at every point by fitting a surface through the point and its 50 neighboring points. Since we are most interested in the z-axis, which corresponds to the camera's point-of-view, we ignore the x- and y-position and normal information. Using the z-position, surface normal in the z-direction, and the local curvature, we apply the same 3-strip partitioning and 15-bin fuzzy histogram to yield a total of $3 \times 15 \times 3 = 135$ depth features.

In order to derive more geometric information from the point cloud, we also calculate the Fast Point Feature Histogram (FPFH) [15] signature for every point. FPFH are informative pose-invariant local features that represent the underlying surface model properties at each point. They are computed based on certain geometric relations between a point and its neighbors. We calculate a 33-bin FPFH for each pixel and the FPFH signature for a rectangle is defined as the sum of FPFH from all pixels within the rectangle.

VI. EXPERIMENTS

A. System Overview

To complete a physical grasp with an industrial robot arm, our system is divided into two parts: offline training and online testing. In offline training, a rank model is learned from a training dataset using SVM-Rank. In online testing, the robot searches an image of the target object for the best grasping rectangle based on the learned model. After the optimal rectangle is found the arm is moved to the predicted location, where it executes the grasp to lift the object.

Fig. 3: Objects used for training.

We performed robotic experiments on a 6-DOF Adept Viper s850 arm mounted with a Microsoft Kinect sensor. The arm has a reach of 105 cm. Its estimated repeatability with grippers is 0.1 mm. The Adept Viper is an industrial arm and has no force or tactile feedback. The kinect sensor that used to perceive point-clouds has a resolution of 640×480 for the depth image and an operational range of 0.8 m to 3.5 m. The Kinect sensor-arm calibration was accurate up to an average of 3 mm.

To build a training dataset, we collected 150 images (with color and depth information on each pixel) of various static objects using the Kinect sensor. The complete set of training objects is shown in Fig. 3. In the training data, every object was placed in various orientations and locations. During the test, objects were also oriented randomly. In each image, we manually labeled 3 good grasping rectangles and randomly generated 5 bad rectangles for each of the two grippers based on their individual abilities.

In online testing, we use two metrics to evaluate a grasp: 1) **Prediction Correctness**, where a predicted rectangle is evaluated by human recognition without executing a physical grasp; and 2) **Grasp and Hold Testing**, where a rectangle is considered a valid grasp if the object can be successfully gripped and held at that place for longer than 15 seconds.

B. Robotic Experiments and Discussion

In robotic experiments, a total of 23 objects were tested for grasping and each object was tested three times at random configurations. The outcome of these tests is shown in Table I. In order to better analyze the results, we divided the objects into five qualitative categories: 1) big and stable; 2) flat and stable; 3) small and stable; 4) unstable; 5) deformable. An object is stable if it will not tip when subjected to any vertically applied force. For example, a mug placed on its side is stable, but a mug placed vertically is not because it will fall over if a vertical force is applied to the handle. To define small/big, we use the size of the jamming gripper as our standard (i.e. an object is small if it can be

TABLE I: Robotic experimental results. Objects that are not graspable for a certain gripper are marked by dash in the corresponding cells.

Category	Object	Jamming Gripper				Parallel Gripper	
		Learning		Baseline: Centroid		Learning	
		Pred(%)	G/H(%)	Pred(%)	G/H(%)	Pred(%)	G/H(%)
Big and Stable	Martini glass (horizontal)	66.7	66.7	33.3	33.3	100	100
	Screw driver	100	100	100	100	100	100
	Brush	100	66.7	66.7	66.7	100	100
	Coffee mug	100	100	0	0	100	100
	Telephone handle	100	66.7	66.7	66.7	-	-
	Lid (upside down)	100	100	100	100	33.3	0
	Toy gun with cord	100	100	0	0	-	-
	AVERAGE	**95.2**	**85.7**	**52.4**	**52.4**	**86.7**	**80.0**
Small and Stable	Battery	100	100	100	100	100	66.7
	Mini-sculpture	100	100	100	100	-	-
	Lens cover	100	100	100	100	-	-
	Charger with cable	66.7	66.7	0	0	100	100
	AVERAGE	**91.7**	**91.7**	**75**	**75**	**100**	**83.4**
Flat	Window wiper	100	66.7	100	66.7	100	100
	Pliers	-	-	-	-	100	100
	Pedal	66.7	66.7	100	100	100	100
	Pen	100	100	100	100	66.7	66.7
	AVERAGE	**88.9**	**77.8**	**100**	**88.9**	**91.7**	**91.7**
Unstable	Tea cup	-	-	-	-	100	100
	Lid	-	-	-	-	100	66.7
	Bowl	-	-	-	-	100	100
	Martini glass (vertical)	-	-	-	-	100	100
	AVERAGE	-	-	-	-	**100**	**91.7**
Deformable	Plastic tongs	100	66.7	100	33.3	100	100
	Gloves	-	-	-	-	100	100
	Shoe	-	-	-	-	100	100
	Purse	-	-	-	-	100	100
	AVERAGE	**100**	**66.7**	**100**	**33.3**	**100**	**100**

fully enveloped by the jamming gripper). Deformable objects are also categorized with respect to the jamming gripper (i.e. to be considered deformable, objects must bend or change shape when the jamming gripper is pressed against them). We will analyze the performance of the grippers on each category in detail.

Learning algorithm vs. heuristic method. The primary goal in these experiments was to demonstrate that our algorithm can identify proper grasps for the jamming gripper. We compare our learning algorithm with a heuristic baseline method (which we call 'centroid') that always grips the center of the object. In detail, we subtract background first to get an approximate region of the object, and then use the centroid of these pixels as the grasping point. Although this simple rule is effective for small objects, it fails when the centroid is located off of the object, or is in some place poorly suited for gripping (such as a phone charger with a long cable). Table I shows the comparison. Snapshots of the jamming gripper grasping objects are shown in Fig. 4

We can see that our algorithm outperforms the 'centroid' method with an average increase in success rate of 18%. For simple-shape objects, such as a pen or a screw driver, the center is usually designed to be a good grasping point. Also for small and stable objects, almost any place on the object is a proper grasp for a jamming gripper. Therefore, both algorithms perform well in these cases. However, for the 'charger with cable' example, the centroid method failed every time because the center was either on the cable or off the object. Our algorithm on the other hand predicted

Fig. 4: The universal jamming gripper grasping different objects.

Fig. 5: The traditional two-jaw gripper that is used to compare with the universal jamming gripper.

only one incorrect rectangle in this case. Beyond this, both methods fail at picking up some items because they are outside the capabilities of the gripper. For example, for unstable objects, the jamming gripper is not always able to pick them up even with manual control. Even if a flat object is graspable, the sensitivity of its point cloud (the depth of the object is very similar with the background and thus almost invisible) can affect our algorithm. Under this circumstance, image-based features are more significant than depth-based features in the score function. Consequently, the algorithm tends to find regions with more changes in color, usually edges of the object, which are sometimes suboptimal. Thus for flat objects, the centroid method sometimes performs better than our learning algorithm.

A special explanation is required for the performance of the jamming gripper on the V-shape plastic tongs. The best grasping position for this item is on its corner, although any location on its legs would seem like a reasonable grasp point. However, away from the corner the legs bend under the pressure of the gripper, leading to a failed grip. This is why the prediction correctness of both algorithms is 100% for the tongs, but successful rate for the physical test is low.

In summary, for stable and non-flat objects that are graspable by the jamming gripper, our algorithm can find proper grasp for the gripper with high reliability. This represents the first time a jamming gripper has successfully executed autonomous closed-loop grasping, and with an average increase in success rate of 18% over a heuristic method.

Grasping with jamming and parallel grippers. To explore the versatility of our learning approach, we also tested grasping the same set of objects with a parallel gripper with two jaws (see Fig. 5). We used the same training data to learn the model for this gripper, but with different labeled grasping rectangles. This is because the good grasps are different for the two grippers. Unlike the jamming gripper, the parallel gripper's orientation would largely influence grasps, so the 'centroid' method, where no orientation is predicted, was not used for comparison. The results are shown in Table I. For stable objects such as a pen, our algorithm could not always find a correct orientation, and some other failures were caused by the limited opening width of the parallel gripper. For these objects, the jamming gripper performs better. Some objects we found the parallel plate gripper could not grasp were: telephone handles, mini-sculptures, and a round lens cover.

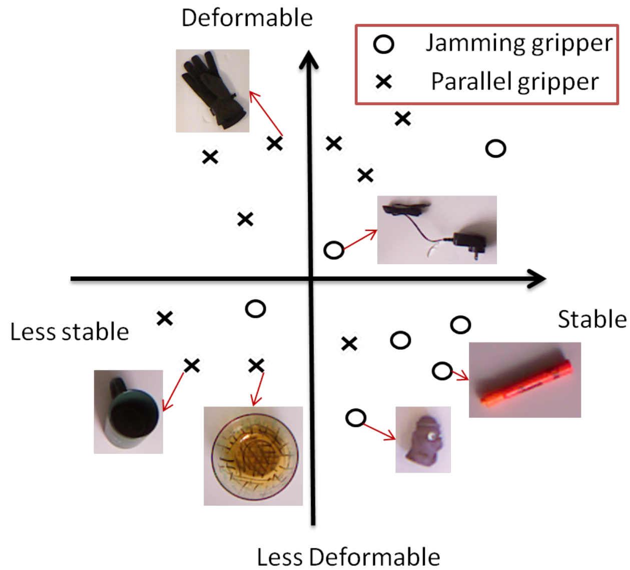

Fig. 6: The preferred gripper for various types of objects. The x-axis stands for stability of the object and the y-axis stands for deformability. The coordinate is only for demonstration, not strictly defined.

One advantage of the parallel gripper is that it is less affected by an object's stability or deformability. So for the parallel gripper, unstable and deformable objects are usually graspable and thus the accuracy on these objects is high. For flat objects as well, the success rate of the parallel gripper is also higher than the jamming gripper. This is mostly because the two stiff parallel plates can provide enough friction (even if the contact is of small size) to hold a flat object. Based on these experimental results, Fig. 6 qualitatively demonstrates the preferred gripper for different objects.

VII. CONCLUSION

In this paper we have demonstrated the first successful execution of autonomous closed-loop grasping with a jamming gripper, through the application of a hardware agnostic learning algorithm that uses RGBD sensor data of the target object but does not require any physical model of the gripper itself. With this learning algorithm we were able to achieve an average increase in success rate of 18% over a simple heuristic gripping method, and were also able to successfully predict grasps on objects not included in the training set. Because the algorithm does not require a physical model of the gripper, we were further able to directly apply the same learning approach to a more traditional two-jawed parallel plate gripper. Our algorithm successfully predicted correct but different grasps for both grippers, and enabled us to compare the types of grasping scenarios that each gripper was best suited for.

In the future, a similar approach could potentially be applied to many other kinds of grippers, and may be especially useful for other soft, flexible, or under-actuated grippers that are difficult to model. If multiple grippers are available (such as on a double-armed robot), a future extension of our hardware agnostic algorithm could be to include a gripper selection feature that predicts grasps and an additional confidence value for each known gripper.

REFERENCES

[1] V. Nguyen, "Constructing stable force-closure grasps," in *ACM Fall joint computer conf*, 1986.

[2] J. Ponce, D. Stam, and B. Faverjon, "On computing two-finger force-closure grasps of curved 2D objects," *IJRR*, vol. 12, no. 3, p. 263, 1993.

[3] K. Lakshminarayana, "Mechanics of form closure," *ASME paper*, p. 32, 1978.

[4] A. Miller and P. K. Allen, "Graspit!: A versatile simulator for robotic grasping," in *IEEE Robotics and Automation Magazine*, vol. 11, no. 4, 2004.

[5] K. Hsiao, S. Chitta, M. Ciocarlie, and E. Jones, "Contact-reactive grasping of objects with partial shape information," in *IROS*, 2010.

[6] K. Hsiao, P. Nangeroni, M. Huber, A. Saxena, and A. Y. Ng, "Reactive grasping using optical proximity sensors," in *ICRA*, 2009.

[7] E. Brown, N. Rodenberg, J. Amend, A. Mozeika, E. Steltz, M. Zakin, H. Lipson, and H. Jaeger, "Universal robotic gripper based on the jamming of granular material," *Proc. Natl. Acad. Sci.*, vol. 107, pp. 18 809–18 814, 2010.

[8] J. Amend, Jr., E. Brown, N. Rodenberg, H. Jaeger, and H. Lipson, "A positive pressure universal gripper based on the jamming of granular material," *IEEE Transactions on Robotics*, to appear 2012.

[9] A. Saxena, J. Driemeyer, J. Kearns, and A. Y. Ng, "Robotic grasping of novel objects," in *NIPS*, 2006.

[10] A. Saxena, J. Driemeyer, J. Kearns, C. Osondu, and A. Y. Ng, "Learning to grasp novel objects using vision," in *ISER*, 2006.

[11] A. Saxena, J. Driemeyer, and A. Ng, "Robotic grasping of novel objects using vision," in *IJRR*, vol. 27, no. 2, 2008, p. 157.

[12] Q. V. Le, D. Kamm, A. Kara, and A. Y. Ng, "Learning to grasp objects with multiple contact points," in *ICRA*, 2010.

[13] Y. Jiang, S. Moseson, and A. Saxena, "Efficient grasping from rgbd images: Learning using a new rectangle representation," in *ICRA*, 2011.

[14] T. Joachims, "Optimizing search engines using clickthrough data," in *SIGKDD*, 2002.

[15] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *ICRA*, 2009, pp. 1848–1853.

[16] A. Bicchi and V. Kumar, "Robotic grasping and contact: a review," in *ICRA*, 2000.

[17] M. T. Mason and J. K. Salisbury, "Manipulator grasping and pushing operations," in *Robot Hands and the Mechanics of Manipulation*. Cambridge, MA: The MIT Press, 1985.

[18] K. Shimoga, "Robot grasp synthesis: a survey," *IJRR*, vol. 15, pp. 230–266, 1996.

[19] N. Niparnan and A. Sudsang, "Fast computation of 4-fingered force-closure grasps from surface points," in *IROS*, vol. 4. IEEE, 2004, pp. 3692–3697.

[20] K. Huebner, K. Welke, M. Przybylski, N. Vahrenkamp, T. Asfour, D. Kragic, and R. Dillmann, "Grasping known objects with humanoid robots: A box-based approach," in *ICRA*. IEEE, 2009, pp. 1–6.

[21] J. Glover, D. Rus, and N. Roy, "Probabilistic models of object geometry for grasp planning," *Robotics Science and Systems IV*, p. 278, 2009.

[22] N. Pollard and V. Zordan, "Physically based grasping control from example," in *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*. ACM, 2005, pp. 311–318.

[23] M. Hueser, T. Baier, and J. Zhang, "Learning of demonstrated grasping skills by stereoscopic tracking of human hand configuration," in *ICRA*, 2006.

[24] A. T. Miller, S. Knoop, P. K. Allen, and H. I. Christensen, "Automatic grasp planning using shape primitives," in *ICRA*, 2003.

[25] J. H. Piater, "Learning visual features to predict hand orientations," in *ICML Workshop on Machine Learning of Spatial Knowledge*, 2002.

[26] A. Morales, P. J. Sanz, and A. P. del Pobil, "Vision-based computation of three-finger grasps on unknown planar objects," in *Int' Robots Sys Conf*, 2002.

[27] D. Bowers and R. Lumia, "Manipulation of unmodeled objects using intelligent grasping schemes," *IEEE Trans Fuzzy Sys*, vol. 11, no. 3, 2003.

[28] A. Saxena, J. Driemeyer, and A. Ng, "Learning 3-d object orientation from images," in *ICRA*, 2009.

[29] A. Saxena, L. Wong, and A. Y. Ng, "Learning grasp strategies with partial shape information," in *AAAI*, 2008.

[30] D. Rao, Q. Le, T. Phoka, M. Quigley, A. Sudsang, and A. Ng, "Grasping Novel Objects with Depth Segmentation," in *IROS*, 2010.

[31] Y. Jiang, C. Zheng, M. Lim, and A. Saxena, "Learning to place new objects," in *ICRA*, 2012.

[32] Y. Jiang, M. Lim, C. Zheng, and A. Saxena, "Learning to place new objects in a scene," in *IJRR*, 2012.

[33] E. Klingbeil, A. Saxena, and A. Ng, "Learning to open new doors," in *IROS*, 2010.

[34] H. Warnecke and I. Schmidt, "Flexible gripper for handling systems: design possibilities and experiences," *Int. J. of Prod. Res.*, vol. 18, pp. 525–537, 1980.

[35] H. Choi and M. Koc, "Design and feasibility tests of a flexible gripper based on inflatable rubber pockets," *Int. J. Mach. Tools & Manuf.*, vol. 46, pp. 1350–1361, 2006.

[36] D. Simpson, "Gripping surfaces for artificial hands," *Hand*, vol. 3, pp. 12–14, 1971.

[37] I. Schmidt, "Flexible moulding jaws for grippers," *Ind. Robot*, vol. 5, pp. 24–26, 1978.

[38] A. Perovskii, "Universal grippers for industrial robots," *Rus. Eng. J.*, vol. 60, pp. 9–11, 1980.

[39] T. Reinmuller and H. Weissmantel, "A shape adaptive gripper finger for robots," *Proc. Int. Symp. on Ind. Robots*, pp. 241–250, 1988.

[40] T. Majmudar, M. Sperl, S. Luding, and R. Behringer, "Jamming transition in granular systems," *Phys. Rev. Lett.*, vol. 98, p. 058001, 2007.

[41] A. Liu and S. Nagel, "Jamming is not just cool any more," *Nature*, vol. 396, p. 2122, 1998.

[42] M. Cates, J. Wittmer, J. Bouchaud, and P. Claudin, "Jamming, force chains, and fragile matter," *Phys. Rev. Lett.*, vol. 81, pp. 1841–1844, 1998.

[43] A. Liu and S. Nagel, *Jamming and rheology: constrained dynamics on microscopic and macroscopic scales*. London: Taylor & Francis, 2001.

[44] C. O'Hern, L. Silbert, A. Liu, and S. Nagel, "Jamming at zero temperature and zero applied stress: the epitome of disorder," *Phys. Rev. E*, vol. 68, p. 011306, 2003.

[45] E. Corwin, H. Jaeger, and S. Nagel, "Structural signature of jamming in granular media," *Nature*, vol. 435, pp. 1075–1078, 2005.

[46] D. Wood, *Soil behavior and critical state soil mechanics*. Cambridge, England: Cambridge University Press, 1990.

[47] P. KaewTraKulPong and R. Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection," in *Proc. European Workshop Advanced Video Based Surveillance Systems*, vol. 1, no. 3. Citeseer, 2001.