Learning to Grasp Novel Objects using Vision

Ashutosh Saxena, Justin Driemeyer, Justin Kearns, Chioma Osondu, Andrew Y. Ng

{asaxena,jdriemeyer,jkearns,cosondu,ang}@cs.stanford.edu
Computer Science Department
Stanford University, Stanford, CA 94305

Summary. We consider the problem of grasping novel objects, specifically, ones that are being seen for the first time through vision. We present a learning algorithm which predicts, as a function of the images, the position at which to grasp the object. This is done without building or requiring a 3-d model of the object. Our algorithm is trained via supervised learning, using synthetic images for the training set. Using our robotic arm, we successfully demonstrate this approach by grasping a variety of differently shaped objects, such as duct tape, markers, mugs, pens, wine glasses, knife-cutters, jugs, keys, toothbrushes, books, and others, including many object types not seen in the training set.

1 Introduction

If we are seeing a novel object for the first time through a vision system, how can we autonomously grasp the object? In this paper, we address the problem of grasping non-deformable objects, including ones not seen before and that the robot is perceiving for the first time through a web-camera.

Modern-day robots can be carefully hand-programmed or "scripted" to carry out amazing manipulation tasks, from using tools to assemble complex machinery, to balancing a spinning top on the edge of a sword [15]. However, fully autonomous grasping of a previously unknown object still remains a challenging problem. If the object was previously known, or if we are able to obtain a full 3-d model of it, then various approaches, for example ones based on friction cones [5], pre-stored primitives [7], or other algorithms can be applied. However, in practical scenarios it is generally very difficult to obtain an accurate 3-d reconstruction of an object that we are seeing for the first time through vision.[1]

In this paper, we show that even without building a 3-d model of the object to be grasped, it is possible to identify a good grasp using learning algorithms. Specifically,

[1] This is particularly true if we have only a single camera. But for objects without texture, even a stereo system would work poorly, and be able to reconstruct only the visible portions of the object. Finally, even if we try to "engineer" the problem away and use a laser (or active stereo) to estimate depths, we would still have only a 3-d reconstruction of the front face of the object.

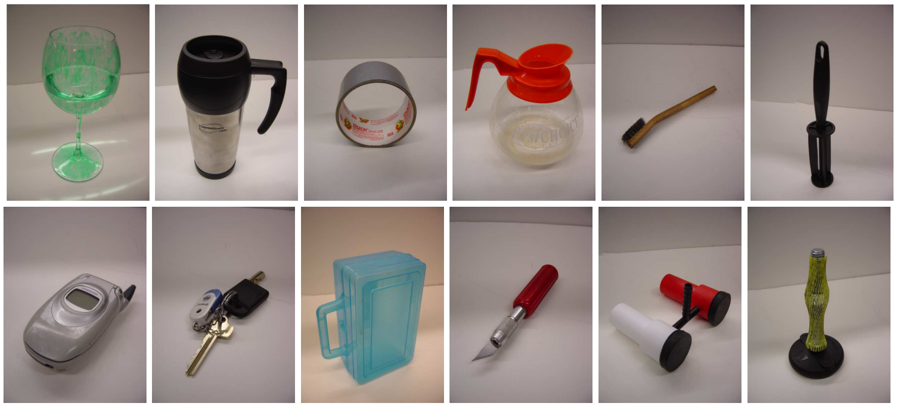

Fig. 1. Some real objects on which the grasping algorithm was tested.

there are certain visual features that indicate good grasps, and that remain consistent across many different objects. For example: jugs, cups, and mugs have handles; objects such as screwdrivers, toothbrushes, etc. are all long objects that can be grasped roughly at their midpoint; and so on. Given only a quick glance at almost any rigid object, most primates can quickly choose a grasp to pick it up; our work represents a first step towards designing a vision grasping algorithm which can do the same. We also take inspiration from Castiello [3], who showed that for commonly used objects, cognitive cues and prior knowledge are used in visually guided grasping by primates.

In prior work, a few others have also applied learning to robotic grasping. [1] For example, Pelossof et al. [9] used a supervised learning algorithm to learn grasps, for settings where a full 3-d model of the object is known. Kaelbling and Lozano-Perez (pers. comm.) also apply learning to grasping, but again assuming a fully known 3-d model of the object. Piater described an algorithm [10] to position single fingers given a top-down view of an object, but considered only very simple objects (specifically, square, triangle and round "blocks"). Platt et al. [11, 12] learned to sequence together manipulation gaits, but again assumed a specific, known, object.

To pick up an object, we need to identify the grasp—more formally, a position and configuration for the end-effector. This paper focuses on the task of grasp identification, and thus we will consider only objects that can be picked up without performing complex manipulation,[2] and that are commonly found in an office or household environment, e.g., toothbrushes, pens, books, mugs, martini glasses, jugs, keys, duct tape rolls, markers. (Fig. 1)

This paper will emphasize grasping previously unknown objects in uncluttered environments (for example, when the objects are placed against a uniform-colored background). The remainder of this paper is structured as follows. Section 2 describes our machine learning approach for grasp identification. Trajectory planning (on our

[2] For example, picking up a heavy book lying flat on table might require a sequence of complex manipulations, such as to first slide it to the edge of the table.

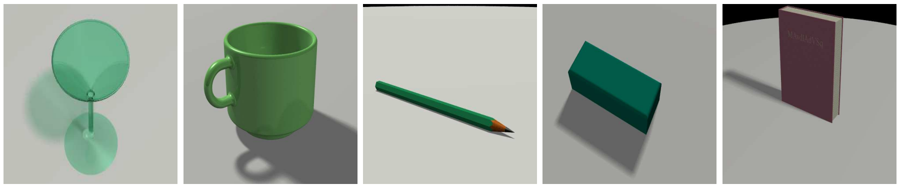

Fig. 2. Synthetic images of the objects used for training.

5 dof arm) is then briefly discussed in Section 3. Section 4 presents our experimental results, and finally Section 5 concludes.

2 Learning the Grasping Point

There are certain visual features that indicate good grasps, and that remain consistent across many different objects. For example: jugs, cups, and mugs have handles; objects like pens, screwdrivers, white-board markers, etc. can be grasped in the center. We propose a learning algorithm that learns to use visual features to identify good grasping points across a large range of objects.

More precisely, we will predict grasp as a function of the image. An image is a projection of the three-dimensional world onto an image plane, which does not have depth information. Therefore, we will predict the 2-d location of the grasp in the image, which corresponds to the projection of the 3-d grasping point into the image plane. We use supervised learning for this task, with synthetic images (generated using computer graphics) as our training data. We then use two (or more) images to triangulate and obtain the 3-d location of the grasp.

2.1 Synthetic Data for Training

Collecting real-world data is cumbersome and manual labeling is prone to errors. Generating perfectly labeled synthetic data is significantly less time-consuming and easier, as compared to real images.

Therefore, we generate synthetic images (Fig. 2), along with labels denoting the correct grasp, using a computer graphics ray tracer.[3] The advantages of using synthetic images are multi-fold [6]. Once a synthetic model for the object has been created, a large number of training examples can be generated with random lighting conditions, camera position and orientation, etc. Additionally, to increase the diversity in our data, we randomized some properties of the object as well, such as color, scale, and text (e.g., on the face of a book). The time-consuming part of synthetic data generation is the manual creation of the numerical models of the objects. However,

[3] Ray tracing [4] is a standard image rendering method in computer graphics. It handles many real-world phenomenon such as multiple specular reflections, texture mapping, soft shadows, smooth curves, and caustics. We used PovRay, an open source ray tracer.

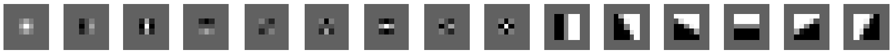

Fig. 3. Examples of different edge and texture filters used to calculate the features.

there are many objects for which models are available on the internet, and can be used with only minor modifications. We generated 2500 examples from synthetic data, comprising instances from five object types. (Fig. 2) Using synthetic data also allows us to generate perfect labels for the training set, i.e., the exact location of a good grasp for each object. In contrast, collecting and manually labeling a comparable-size set of real images would have been extremely time-consuming.

There is a trade-off between the quality of synthetically generated images and the accuracy of the algorithm. The better the quality of the synthetic images and graphical realism, the better the accuracy of the algorithm. Therefore, we use a ray tracer instead of faster, but cruder, openGL style graphics.[4] Ray tracers allow generation of details seen in real images, which are difficult, if not impossible, to generate in simpler graphics implementations. We use these more lifelike images to allow our learned model to become robust to the presence of these phenomena in real images.

2.2 Grasping Point Classification

Given the training set, our algorithm learns to identify grasping regions in the images. More precisely, given the training set, the learning algorithm predicts the 2-d position of the grasp projected into the image plane. The algorithm uses a set of features of the image, which include edges and texture information, applied at various scales. Using these features, we apply logistic regression to decide whether each position in the 2-d image plane corresponds to a valid grasping point.

In detail, the logistic regression algorithm models the probability of a particular patch of the image being a valid grasping point as:

$$p(y = 1|x; w) = \frac{1}{1 + e^{-w^T x}} \tag{1}$$

Here, $w \in \mathbb{R}^{459}$ are the parameters, which are learned by maximum likelihood. The features $x \in \mathbb{R}^{459}$ we use for the patch include edges and texture information, (Fig. 3) applied at three spatial scales, and appended with the filter outputs at the lowest scale for the surrounding patches. (See [13] for more detail on the image features.) Fig. 4 shows some predicted grasps on real images.

2.3 Approximate Triangulation

Given two (or more) images of a new object from different camera positions and the predicted 2-d grasp positions in each image, we need to triangulate to obtain

[4] Michels, Saxena and Ng [6] used synthetic openGL images to learn distances in natural scenes. However, because of the cruder rendering style of openGL graphics, the learning performance sometimes *decreased* with added complexity in the scenes.

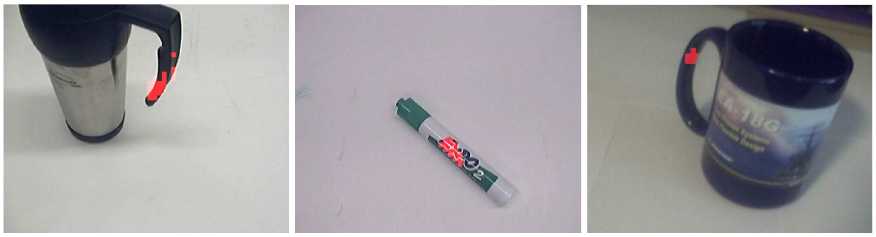

Fig. 4. Grasping point classification. The red points show the predicted valid grasping points.

3-d positions of the grasping points (Fig. 5). Note that we perform triangulation only to identify the 3-d position of the grasp, not for full 3-d reconstruction. Indeed, many of our test objects are textureless or reflective, and 3-d reconstruction using standard stereopsis would have performed poorly on them. We use a triangulation algorithm that is more complex than one based on standard geometric calculations to handle the learning algorithm's output being slightly noisy/uncertain, and to handle the possibility of there being multiple valid grasping points on an object.

We use a counting algorithm for this "triangulation" step.[5] First, we discretize the 3-d space of possible grasping points $P \in \mathbb{R}^3$ into a uniform 50x50x50 grid G. Then, using knowledge of the camera position $C \in \mathbb{R}^3$ and pose, each predicted grasp in the image plane becomes a ray, $R = C + t\hat{r}$, with direction $\hat{r} \in \mathbb{R}^3, ||\hat{r}||_2 = 1$. Here, $t \in \mathbb{R}_+$ represents the distance along the ray from the camera. To account for uncertainty in the prediction, we assume a Gaussian error around this ray with variance $\sigma^2 = \alpha t^2$, which increases with distance t from the camera, because of image projective transformation. Thus, each ray is represented by a cone centered on the ray with Gaussian spread. We then count the response from each Gaussian cone for each grid point. More formally, the response Ψ at a point $p \in G$ is given as

$$\Psi(p) = \sum_{j=1}^{N_I} \sum_{k=1}^{N_j} \exp\left(-\frac{\text{dist}(p, R_{jk})^2}{2\sigma^2}\right) \tag{2}$$

where, N_I is the total number of images, N_j is the total number of rays from the image j, and R_{jk} is the k^{th} ray from the j^{th} image. Using this method, we take the grid cell p^* with the highest total response as the predicted grasping point $p^* = \max_{p \in G} \Psi(p)$.

In reality, due to the ambiguous nature of a grasping point (i.e., the entire handle is equally valid, not just one spot), the grasping point classification algorithm, followed by statistical triangulation, gives high response Ψ for multiple grid cells clustered around the valid grasping point. Therefore, we increase the robustness of our choice of the grasping point by taking into account the spatial structure of the objects.[6] In

[5] We give details on inference of 3-d grasping points using probabilistic models in [14].

[6] To account for this, instead of predicting just the grid cell of highest response, we instead locate all the grid cells in the top 10% in terms of response Ψ. Formally, we define a subset $G' \subset G, G' = \{p \mid \Psi(p) \geq 0.9\,\Psi(p^*)\ \}$. Among G', we identify which $p \in G'$ minimizes the total distance to the other high response grid cells, and predict that as the grasping

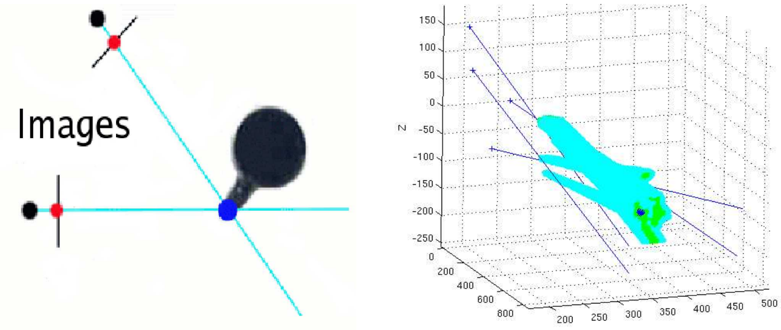

Fig. 5. Statistical Triangulation of the grasping point. (a) A sketch showing rays from two images intersecting at the grasping point (shown in dark blue). (b) A 3-d plot showing multiple Gaussian cones (in light blue) from 3 images, intersecting at the grasping point (shown in dark blue).

our experiments, this significantly increases algorithmic robustness in the face of ambiguity in triangulation.

3 Control

To grasp an object (using our 5-dof arm), we plan a trajectory to take the end-effector to an approach position,[7] and then move the end-effector in a straight line towards the predicted grasping point. Trajectory planning was done in joint angle space. We used linear interpolation between successive steps to the approach position.

We use two classes of grasps: *downward* and *outward*, which arise because of the workspace of the 5-dof arm (Fig. 6). A "downward" grasp is for objects that are close to the base of the arm, and which the arm can reach in a downward direction. An "outward" grasp is for objects further away from the base, which the arm is unable to reach in a downward direction. To choose the class of the grasp, we scan the workspace of the arm and determine which region contains the object.[8]

4 Experiments

4.1 Hardware Setup

We used the STAIR (STanford AI Robot, see Fig. 7) robot built at Stanford University. This platform is equipped with a robotic arm mounted on a mobile platform, along

point $p^{**} = \min_{p \in G'} \sum_{q \in G'} ||p - q||_1$. This incorporates the fact that identified grasping regions should be clustered together.

[7] The approach position is defined to be a point a fixed distance away from the predicted grasp point.

[8] We determine the position of objects in the picture of the robot workspace by thresholding the saturation channel of HSV (Hue-Saturation-Value) of the image. We use this position to determine the region in which the object lies.

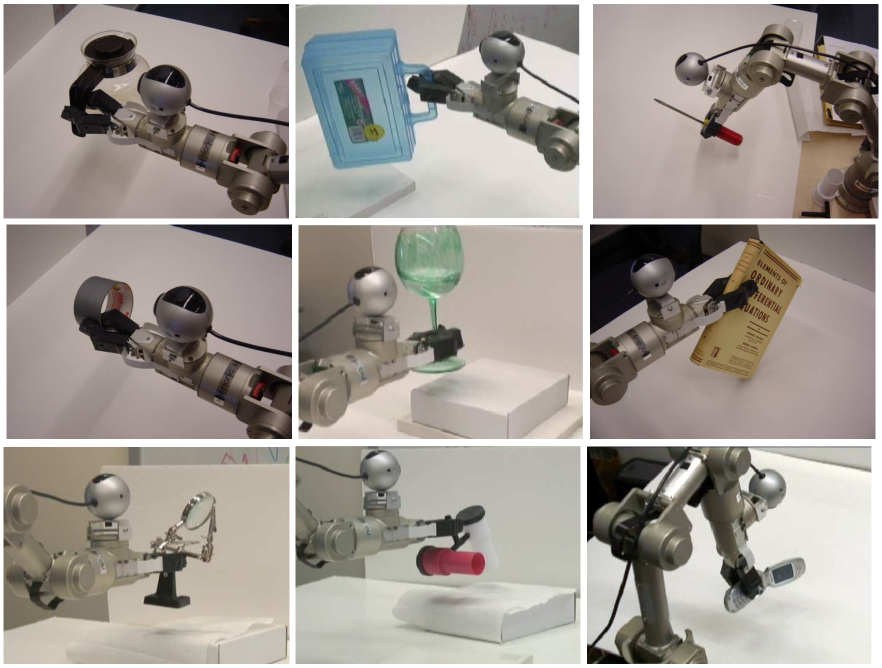

Fig. 6. The robotic arm picking up various objects: jug, box, screwdriver, duct-tape, wine glass, book, a chip-holder, powerhorn, and cellphone.

with other equipment such as cameras, microphones, etc. The long-term goal of the STAIR project is to create a robot that can navigate home and office environments, pick up and interact with objects and tools (including carrying out more complex tasks such as unloading a dishwasher), and intelligently converse with and help people in these environments. Clearly, the ability to grasp a novel object represents an interesting and necessary step towards these goals.

The robotic arm on STAIR is a light 4 kg, 5-dof arm (Katana [8]) equipped with a parallel plate gripper. It holds a payload of 500g, and has a horizontal reach of 62cm, and a vertical reach of 79cm. The positioning accuracy of the arm is ±1 mm. It is a position controlled arm, i.e., it requires specification of joint locations instead of torques. Our vision system uses a low-quality webcam mounted near the end-effector.

4.2 Results and Discussion

We first tested the algorithm for its predictive capability on synthetic images not in the training set. The average classification accuracy was 94.2%, although the accuracy in predicting a 3-d grasping point was higher than the classification accuracy, because 3-d triangulation “fixes” some errors in the classification step.

Next, we tested the algorithm on the STAIR robot. The task was to use input from a web-camera, mounted on the robot, to pick up an object placed in front of the robot against a white background. The parameters of the vision algorithm were trained

Table 1. Average absolute error in locating the grasping point for different objects, as well as success rate in grasping objects using our robotic arm. (Although training was done on synthetic images, all testing was done on the robotic arm and real objects.)

Objects similar to ones in the training set

Tested on	Mean Error (cm)	Grasp-rate
Mugs	2.8	75%
Pens	0.9	100%
Wine Glass	1.1	100%
Books	2.9	75%
Eraser/Cellphone	1.6	100%
Overall	1.9	90%

Novel Objects

Tested on	Mean Error (cm)	Grasp-rate
Keys/Markers	1.2	100%
Toothbrush/Cutter/Screwdriver	1.1	100%
Jug	1.7	75%
Powerhorn	3.5	50%
Duct Tape	1.8	100%
Coiled Wire	1.4	100%
Overall	1.8	87.5%

from synthetic images of a small set of objects, namely books, martini glasses, whiteboard erasers, coffee mugs, tea cups and pencils. We performed experiments on coffee mugs, wine glasses, pencils, books, and erasers—but all of different dimensions and appearance than the ones in the training set—as well as a large set of novel objects, such as duct tape rolls, markers, a translucent box, jugs, knife-cutters, a cellphone, pens, keys, screwdrivers, a stapler, toothbrushes, a thick coil of wire, a strangely shaped power horn, etc. (Fig. 1 and 6)

In extensive experiments, the algorithm for predicting grasps in images appeared to generalize very well. Despite being tested on images of real (rather than synthetic) objects, including many very different from ones in the training set, it was usually able to identify correct grasping points. We note that test error (in terms of average error in predicting a good grasping point) on the real images was only somewhat higher than the error on synthetic images, showing that the algorithm trained on synthetic images transfers well to real images. (Over all 5 object types used in the synthetic data, average absolute error was 0.8cm[9] in the synthetic images; and over all the 11 real test objects, average error was 1.8cm.) For comparison, neonate humans can grasp simple objects with an average accuracy of 1.5 cm. [2]

[9] Units based on typical size of real world objects represented by the synthetic images (e.g., a typical mug is 12 cm high, etc.)

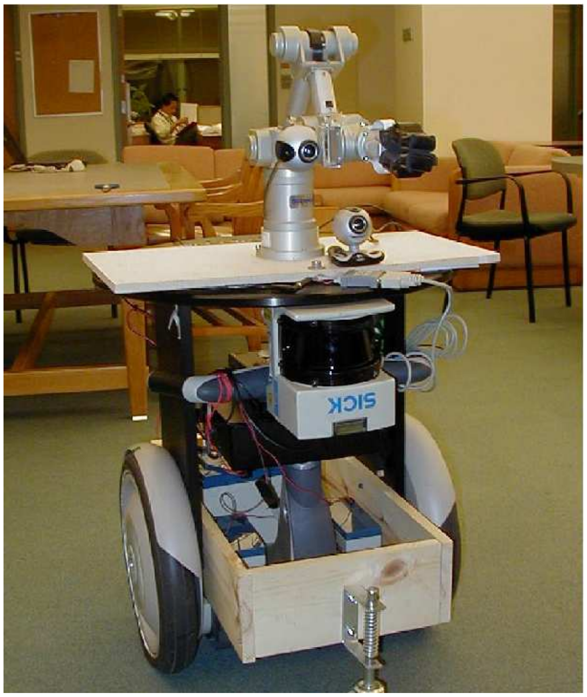

Fig. 7. The STAIR robot. The 5-dof arm was used to pick up the objects.

Table 1 shows the errors in actual grasping points that we obtained on the real dataset. The table presents results separately for objects which are similar to those we trained on (e.g., coffee mugs) and those which were very dissimilar to the training objects (e.g., duct tape). In addition to reporting errors in grasp positions, we also report the grasp-rate, i.e., the fraction of times the robotic arm was able to physically pick up the object (out of 4 trials). On average, the robot succeeded in picking up a novel object 87.5% of the time.

For simple objects such as cellphones, wine glasses, keys, toothbrushes, etc., the algorithm performed perfectly (100% grasp-rate). However, objects such as mugs and jugs allow only a narrow trajectory of approach; as a result, a minor error in grasping point prediction can cause the arm to hit and move the object, resulting in failure to grasp, and thus a lower overall success rate. We believe that these problems can be solved with better control strategies using haptic feedback. Some of the failures can also be attributed to the fixed gripper width used across all objects; this can be solved by learning how much the gripper should open. Videos of the arm picking up various objects are available at

http://ai.stanford.edu/∼asaxena/learninggrasp/

In many instances, the algorithm was able to pick up completely novel objects (strangely shaped power-horn, duct-tape, etc.; see Fig. 6) by identifying grasping points. Perceiving a transparent wine glass is a difficult problem for standard vision (e.g., stereopsis) algorithms because of reflections, etc. However, as shown in Table 1, our algorithm successfully picked it up 100% of the time. The same rate of success holds even if the glass is 2/3 filled with water.

5 Conclusions

We described a machine learning algorithm for identifying a grasping point on a previously unknown object that a robot is perceiving for the first time using vision. Our algorithm does not require (and nor does it build) a 3-d model of the object, and was applied to grasping a number of novel objects using our robotic arm.

Acknowledgment

We thank Anya Petrovskaya and Jimmy Zhang for help with the robotic arm control driver software. This work was supported by the DARPA transfer learning program under contract number FA8750-05-2-0249.

References

1. A. Bicchi and V. Kumar. Robotic grasping and contact: a review. In *ICRA*, 2000.
2. T. Bower, J. Broughton, and M. Moore. Demonstration of intention in the reaching behaviour of neonate humans. *Nature*, 228:679–681, 1970.
3. U. Castiello. The neuroscience of grasping. *Nature Reviews Neuroscience*, 6, 2005.
4. A. S. Glassner. *An Introduction to Ray Tracing*. Morgan Kaufmann Publishers, Inc., San Francisco, 1989.
5. M. T. Mason and J. K. Salisbury. Manipulator grasping and pushing operations. In *Robot Hands and the Mechanics of Manipulation*. The MIT Press, Cambridge, MA, 1985.
6. J. Michels, A. Saxena, and A. Y. Ng. High speed obstacle avoidance using monocular vision and reinforcement learning. In *ICML*, 2005.
7. A. Miller, S. Knoop, H. Christensen, and P. Allen. Automatic grasp planning using shape primitives. In *ICRA*, 2003.
8. Neuronics. Katana user manual. *http://www.neuronics.ch/*, 2004.
9. R. Pelossof, A. Miller, P. Allen, and T. Jebara. An svm learning approach to robotic grasping. In *ICRA*, 2004.
10. J. H. Piater. Learning visual features to predict hand orientations. In *ICML Workshop on Machine Learning of Spatial Knowledge*, 2000.
11. R. Platt, A. H. Fagg, and R. Grupen. Manipulation gaits: Sequences of grasp control tasks. In *ICRA*, 2004.
12. R. Platt, A. H. Fagg, and R. Grupen. Reusing schematic grasping policies. In *IEEE-RAS International Conference on Humanoid Robots, Tsukuba, Japan*, 2005.
13. A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In *NIPS 18*, 2005.
14. A. Saxena, J. Driemeyer, J. Kearns, and A. Y. Ng. Robotic grasping of novel objects. To appear in *NIPS*, 2006.
15. T. Shin-ichi and M. Satoshi. Living and working with robots. *Nipponia*, 2000.