Autonomously Acquiring Instance-Based Object Models from Experience

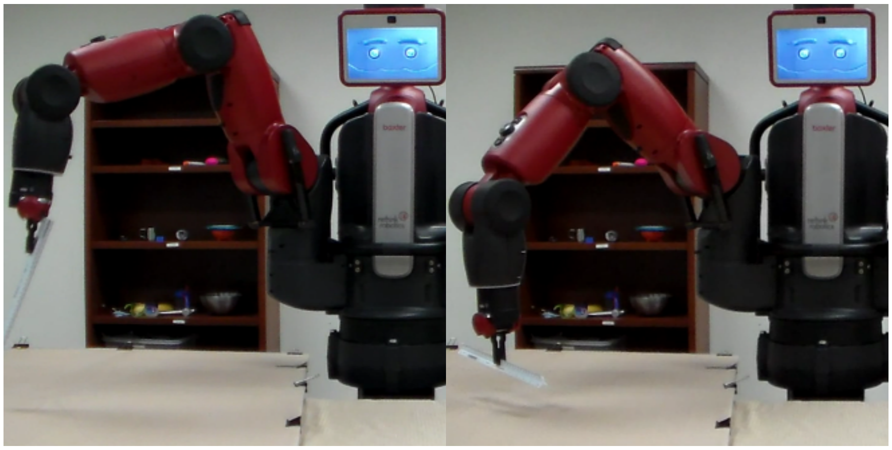
John Oberlin and Stefanie Tellex

Abstract A key aim of current research is to create robots that can reliably manipulate objects. However, in many applications, general-purpose object detection or manipulation is not required: the robot would be useful if it could recognize, localize, and manipulate the relatively small set of specific objects most important in that application, but do so with very high reliability. Instance-based approaches can achieve this high reliability but to work well, they require large amounts of data about the objects that are being manipulated. The first contribution of this paper is a system that automates this data collection using a robot. When the robot encounters a novel object, it collects data that enables it to detect the object, estimate its pose, and grasp it. However for some objects, information needed to infer a successful grasp is not visible to the robot's sensors; for example, a heavy object might need to be grasped in the middle or else it will twist out of the robot's gripper. The second contribution of this paper is an approach that allows a robot to identify the best grasp point by attempting to pick up the object and tracking its successes and failures. Because the number of grasp points is very large, we formalize grasping as an N-armed bandit problem and define a new algorithm for best arm identification in budgeted bandits that enables the robot to quickly find an arm corresponding to a good grasp without pulling all the arms. We demonstrate that a stock Baxter robot with no additional sensing can autonomously acquire models for a wide variety of objects and use the models to detect, classify, and manipulate the objects. Additionally, we show that our adaptation step significantly improves accuracy over a non-adaptive system, enabling a robot to improve its pick success rate from 55% to 75% on a collection of 30 household objects. Our instance-based approach exploits the robot's ability to collect its own training data, enabling experience with the object to directly improve the robot's performance during future interactions.

1 Introduction

Robotics will assist us at childcare, help us cook, and provide service to doctors, nurses, and patients in hospitals. Many of these tasks require a robot to robustly perceive and manipulate objects in its environment, yet robust object manipulation remains a challenging problem. Transparent or reflective surfaces that are not visible in IR or RGB make it difficult to infer grasp points [24], while emergent physical

Brown University



(a) Before learning, the ruler slips. (b) After learning, the robot picks it up.

Fig. 1 Before learning, the robot grasps the ruler near the end, and it twists out of the gripper and falls onto the table; after learning, the robot successfully grasps near the ruler's center of mass.

dynamics cause objects to slip out of the robot's gripper; for example, a heavy object might slip to the ground during transport unless the robot grabs it close to the center of mass. Instance-based approaches that focus on specific objects can have higher accuracy but usually require training by a human operator, which is time consuming and can be difficult for a non-expert to perform [15, 19, 20]. Existing approaches for autonomously learning 3D object models often rely on expensive iterative closest point-based methods to localize objects, which are susceptible to local minima and take time to converge [17].

To address this problem, we take an instance-based approach, exploiting the robot's ability to collect its own training data. Although this approach does not generalize to novel objects, it enables experience with the object to directly improve the robot's performance during future interactions, analogous to how mapping an environment improves a robot's ability later to localize itself. After this data collection process is complete, the robot can quickly and reliably manipulate the objects. Our first contribution is an approach that enables a robot to achieve the high accuracy of instance-based methods by autonomously acquiring training data on a per object basis. Our grasping and perception pipeline uses standard computer vision techniques to perform data collection, feature extraction, and training. It uses active visual servoing for localization, and only uses depth information at scan time. Because our camera can move with seven degrees of freedom, the robot collects large quantities of view-based training data, so that straightforward object detection approaches perform with high accuracy. This framework enables a Baxter robot to detect, classify, and manipulate many objects.

However, limitations in sensing and complex physical dynamics cause problems for some objects. Our second contribution addresses these limitations by enabling a robot to learn about an object through exploration and adapt its grasping model

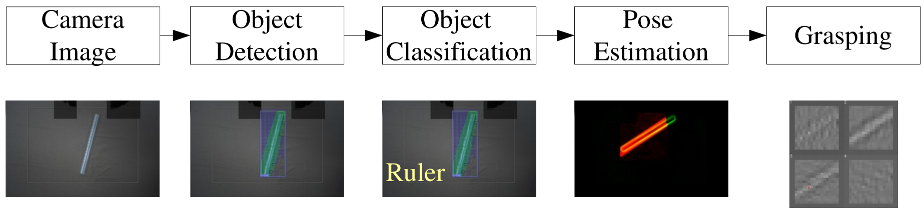


Fig. 2 Results at each phase of the grasping pipeline.

accordingly. We frame the problem of model adaptation as identifying the best arm for an N-armed bandit problem [41] where the robot aims to minimize simple regret after a finite exploration period [3]. Existing algorithms for best arm identification require pulling all the arms as an initialization step [27, 1, 5]; in the case of identifying grasp points, where each grasp takes more than 15 seconds and there are more than 1000 potential arms, this is a prohibitive expense. To avoid pulling all the arms, we present a new algorithm, Prior Confidence Bound, based on Hoeffding races [28]. In our approach, the robot pulls arms in an order determined by a prior, which allows it to try the most promising arms first. It can then autonomously decide when to stop by bounding the confidence in the result. Figure 1 shows the robot's performance before and after training on a ruler; after training it grasps the object in the center, improving the success rate.

Our evaluation demonstrates that our scanning approach enables a Baxter robot with no additional sensing to detect, localize, and pick up a variety of household objects. Further, our adaptation step improves the overall pick success rate from 55% to 75% on our test set of 30 household objects, shown in Figure 5.

2 Grasping System

Our object detection and pose estimation pipeline uses conventional computer vision algorithms in a simple software architecture to achieve a frame rate of about 2Hz for object detection and pose estimation. Object classes consist of specific object instances rather than general object categories. Using instance recognition means we cannot reliably detect categories, such as "mugs," but the system will be much better able to detect, localize, and grasp the specific instances, e.g. particular mugs, for which it does have models.

Our detection pipeline runs on stock Baxter with one additional computer. The pipeline starts with video from the robot's wrist cameras, proposes a small number of candidate object bounding boxes in each frame, and classifies each candidate bounding box as belonging to a previously encountered object class. When the robot moves to attempt a pick, it uses detected bounding boxes and visual servoing to move the arm to a position approximately above the target object. Next, it uses image gradients to servo the arm to a known position and orientation above the object. Because we can know the gripper's position relative to the object, we can reliably collect statistics about the success rate of grasps at specific points on the object.

2.1 Object Detection

The goal of the object detection component is to extract bounding boxes for objects in the environment from a relatively uniform background. The robot uses object detection to identify regions of interest for further processing. The input of the object detection component is an image, I; the output is a set of candidate bounding boxes, B. Our object detection approach uses a modified Canny algorithm which terminates before the usual non-maximal suppression step [4]. We start by converting I to a YCbCr opponent color representation. Then we apply 5×5 Sobel derivative filters [39] to each of the three channels and keep the square gradient magnitude. We take a convex combination of the three channels, where Cb and Cr and weighted the same and more heavily than Y because Y contains more information about shadows and specular information, which adds noise. Finally we downsample, apply the two Canny thresholds, and find connected components. We generate a candidate bounding box for each remaining component by taking the smallest box which contains the component. We throw out boxes which do not contain enough visual data to classify. If a box is contained entirely within another, we discard it.

2.2 Object Classification

The object classification module takes as input a bounding box, B, and outputs a label for that object, c, based on the robot's memory. This label is used to identify the object and look up other information about the object for grasping further down the pipeline. For each object c we wish to classify, we gather a set of example crops E_c which are candidate bounding boxes (derived as above) which contain c. We extract dense SIFT features [22] from all boxes of all classes and use k-means to extract a visual vocabulary of SIFT features [40]. We then construct a Bag of Words feature vector for each image and augment it with a histogram of colors which appear in that image. The augmented feature vector is incorporated into a k-nearest-neighbors model which we use to classify objects at inference [40]. We use kNN because our automated training process allows us to acquire as much high-quality data as necessary to make the model work well, and kNN supports direct matching to this large dataset.

2.3 Pose Estimation

For pose estimation, we require a crop of the image gradient of the object at a specific, known pose. As during the bounding box proposal step, we approximate the gradient using 5×5 Sobel derivative filters [39], but we use a different convex combination of the channels which focuses even less on the Y channel. Camera noise in the color channels is significant. To cope with the noise, we marginalize the gradient estimate over several frames taken from the same location, providing a much cleaner signal which matches more robustly. To estimate pose, we rotate our training image and find the closest match to the image currently recorded from the camera, as detected and localized via the pipeline in Section 2.1 and 2.2. Once the pose is determined, we have enough information to attempt any realizable grasp, but our system focuses on crane grasps.

Lighting changes between scan and pick time can make it difficult to perform image matching. In order to match our template image with the crop observed at pick time, we remove the mean from the two images and L^2 normalize them. Removing the mean provides invariance to bias, and normalizing introduces invariance to scaling. These both help to provide compensation for inadequacies in the lighting.

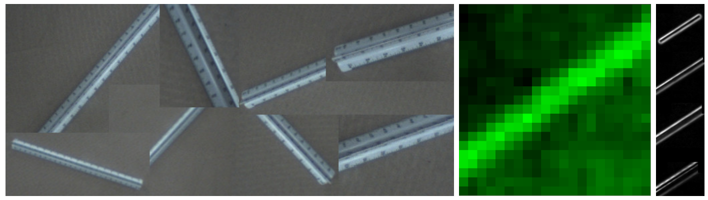
2.4 Grasping

During grasp point identification, we use a model of the gripper to perform inference over a depth map of the object. The grasp model scores each potential grasp according to a linear model of the gripper in order to estimate grasp success. A default algorithm picks the highest-scoring grasp point using hand designed linear filters, but frequently this point is not actually a good grasp, because the object might slip out of the robot's gripper or part of the object may not be visible in IR. The input to this module is the 3D pose of the object, and the output is a grasp point (x, y, θ); at this point we employ only crane grasps rather than full 3D grasping, where θ is the angle which the gripper assumes for the grasp. This approach is not a state-of-the-art but is simple to implement and works well for many objects in practice. In Section 4, we describe how we can improve grasp proposals from experience, which can in principle use any state-of-the-art grasp proposal system as a prior.

3 Autonomously Acquiring Object Models

An object model in our framework consists of the following elements, which the robot autonomously acquires:

- cropped object templates (roughly 200), $t^1 \ldots t^K$
- depth map, D, which consists of a point cloud, $(x, y, z, r, g, b)^{i,j}$.
- cropped gradient templates at different heights, $t_0 \ldots t^M$



(a) (b) (c)

Fig. 3 Autonomously acquired object model. 3(a): Cropped RGB images, 3(b): Depth map, 3(b): Aerial gradient images at four different heights.

The robot collects gradient images by servoing to the center of the extracted bounding box for the object, described in Section 2.1, and then recording a gradient image at several different heights. It records each image for several frames to average away noise from the camera. Gradient images for the ruler appear in Figure 3(c).

Next, it acquires a depth image. Normally, this image could be acquired from an RGB-D sensor such as the Kinect. However, in order to make our approach run on a stock Baxter robot with no additional sensing, we acquire a depth scan using Baxter's IR sensor, turning the arm into a seven degree of freedom, one-pixel depth sensor. After acquiring visual and IR models for the object at different poses of the arm, we acquire view-based object models for detection and classification by moving the camera around the object, extracting bounding boxes from the images, and storing the resulting crops. Figure 3(a) shows RGB images automatically collected for one object in our dataset.

4 Bandit-based Model Adaptation

The formalization we contribute treats grasp learning as an N-armed bandit problem. Formally, the agent is given an N-armed bandit, where each arm pays out 1 with probability μ_i and 0 otherwise. The agent's goal is to identify a good arm (with payout $\geq k$) with probability c (e.g., 95% confidence that this arm is good) as quickly as possible. As soon as it has done this, it should terminate. The agent is also given a prior π on the arms so that it may make informed decisions about which grasps to explore.

4.1 Algorithm

Our algorithm, Prior Confidence Bound, iteratively chooses the arm with the highest observed (or prior) success rate but whose probability of being below k is less than a threshold. It then tries that arm, records the results, and updates its estimate of the probability of success, μ_i. If it is sufficiently certain that the arm's payout is either very good or very bad, it terminates; otherwise, it continues pulling the arm to collect more information. Pseudo-code appears in Algorithm 1. Our algorithm takes as input π, an estimate of the payout of each arm, as well as δ_{accept} and δ_{reject}, parameters controlling how certain it must be to accept or reject an arm. We need to estimate the probability that the true payout probability, μ_i, is greater than the threshold, c, given the observed number of successes and failures:

$$\Pr(\mu_i > k|S,F) \tag{1}$$

We can compute this probability using the law of total probability:

$$\Pr(\mu_i > k|S,F) = 1 - \int_0^k \Pr(\mu_i = \mu|S,F)d\mu \tag{2}$$

We assume a beta distribution on μ:

$$= \int_k^1 \mu^S(1-\mu)^F d\mu \tag{3}$$

This integral is the CDF of the beta distribution, and is called the regularized incomplete beta function [32].

The prior controls both the order that arms are explored and when the algorithm moves on to the next arm. If the prior is optimistic (i.e., overestimates μ_i), the algorithm will more quickly move on to the next arm if it encounters failures, because its empirical estimate of μ_i will be lower than the estimate from the prior of the next arm to pull. If the prior is pessimistic, the algorithm will be more likely to continue pulling an arm even if it encounters failure. By using a prior that incorporates probability estimates, it enables our algorithm to exploit information from the underlying grasp proposal system and make more informed decisions about when to move on.

4.2 Simulation

We simulate our algorithm by creating a sequence of 50 bandits, where each arm i pays out at a rate uniformly sampled between 0 and 1. For algorithms that incorporate prior knowledge, we sample a vector of estimates for each μ_i from a beta distribution with $\alpha = \beta = 1 + e * \mu_i$ where e controls the entropy of the sampling distribution.

To compare to a well-known baseline, we assess the performance of Thompson Sampling [41] in the fixed budget setting, although this algorithm minimizes total regret, including regret during training, rather than simple regret. Second, we compare to a Uniform baseline that pulls every arm equally until the budget is exceeded. This baseline corresponds to the initialization step in UCB or the confidence bound algorithms in Chen et al. [5]. The state-of-the-art CLUCB algorithm from Chen et al. [5] would not have enough pulls to finish this initialization step in our setting. Finally, we show the performance of three versions of Prior Confidence Bound, one

```
PriorConfidenceBound(π, k, δ_accept, δ_reject, maxTries)
Initialize S_0 ... S_n to 0
Initialize F_0 ... F_n to 0
totalTries ← 0
while true do
    totalTries ← totalTries + 1
    Set M_0 ... M_n to S_0/(S_0+F_0) ... S_n/(S_n+F_n)
    j ← bestValidArm;  // set j to the arm with p_below < δ_reject that has
    the highest marginal value
    r ← sample(arm_j)
    if r = 1 then
        S_j ← S_j + 1
    else
        F_j ← F_j + 1
    end
    p_below ← ∫_0^k Pr(μ_j = μ|S_j, F_j)dμ
    p_above ← ∫_k^1 Pr(μ_j = μ|S_j, F_j)dμ
    p_threshold ← ∫_{k−ε}^{k+ε} Pr(μ_j = μ|S_j, F_j)dμ
    if p_above ≥ δ_accept then
        return j ;                                  // accept this arm
    else if p_threshold ≥ δ_accept then
        return j ;                                  // accept this arm
    else if totalTries ≥ maxTries then
        return maxI ; // return the arm with the best marginal value
        out of those that were tried
    else
        pass ;                                      // keep trying
    end
end
```

Algorithm 1: Prior Confidence Bound for Best Arm Identification

with an uninformed prior ($e = 0$, corresponding to Hoeffding races [28]), one quite noisy with $e = 1$(but still informative), the other less noisy $e = 5$).

We run each experiment for 100 trials, and report 95% confidence intervals around the algorithm's simple regret. For Thompson Sampling and Uniform, which always use all trials in their budget, we report performance at each budget level; for Prior Confidence Bound, we report the mean number of trials the algorithm took before halting, also at 95% confidence intervals.

Results appear in Figure 4. Thompson Sampling always uses all trials in its budget and improves performance as larger budgets are available. The Uniform method fails to find the optimal arm because there is not enough information when pulling each arm once. All variants of Prior Confidence Bound outperform these baselines, but as more prior information is incorporated, regret decreases. Even with a completely uninformed prior, bounding the confidence and decided when to stop improves performance over Thompson sampling or a uniform baseline, but the approach realizes significant further improvement with more prior knowledge.

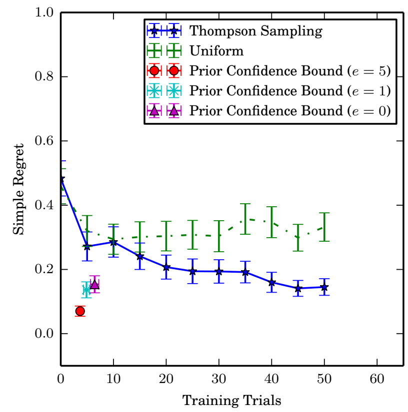


Fig. 4 Results comparing our approach to various baselines in simulation.

Fig. 5 The objects used in our evaluation, sorted from worst performing (left) to best performing (right).

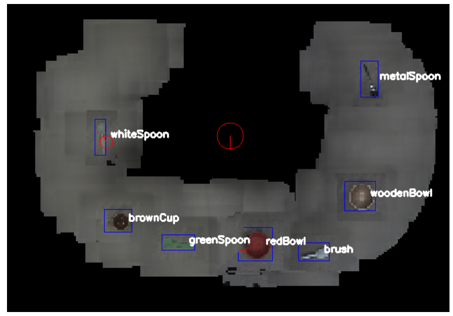
5 Evaluation

The aim of our evaluation is to assess the ability of the system to acquire visual models of objects which are effective for grasping and object detection. We implemented our approach on a Baxter robot; a video showing our training and grasping pipeline is available at `https://www.youtube.com/watch?v=xfH0B3g782Y`.

5.1 Mapping

Mapping assesses the ability of our robot to accurately localize and label objects in a tabletop scene. The robot maps the scene by maintaining a data structure with an

(a) Tabletop scene with objects



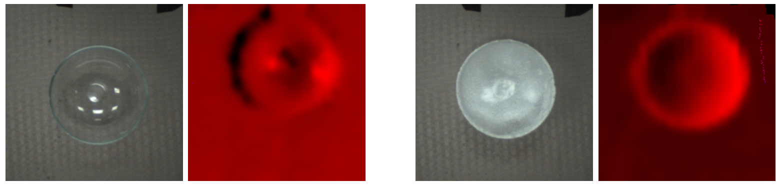
(b) Map created for the scene by scanning with both arms.

Fig. 6 The robot actively scans the table and maps its environment using learned models. (Descriptive object labels are provided by hand.)

entry for each cell in its work space, at approximately 1cm resolution, and recording the last time that cell was observed by the camera. It samples a new cell uniformly from the set of oldest cells, moves to that location, then runs the detection step. If it sees an object, it servos to that object, then adds the object's bounding box and class label to the map. By running object classification directly over the object, we obtain high-accuracy recognition rates, since the robot sees the object from a consistent pose. Figure 6 shows the map created in this way for a tabletop scene. We compute colors for each cell by taking the average of camera pixel colors at that cell, given the current table height.

5.2 *Pick and Place*

The robot acquired visual and RGB-D models for 30 objects using our autonomous learning system. The objects used in our evaluation appear in Figure 5. We manually verified that the scans were accurate, and set the following parameters: height above the object for the IR scan (to approximately 2cm); this height could be acquired automatically by doing a first coarse IR scan following by a second IR scan 2cm above the tallest height, but we set it manually to save time. Additionally we set the height of the arm for the initial servo to acquire the object. After acquiring visual and IR models for the object at different poses of the arm, the robot performed the bandit-based adaptation step using Algorithm 1. The algorithm requires a scoring of candidate grasps, π, which we provided using the linear filter described in Section 2.4. In principle, we could use any state-of-the-art system for proposing grasps in the prior (e.g., [9, 34, 36]); if the proposed grasp is successful, the algorithm will quickly terminate. Otherwise it will continue trying to pick until it finds a successful grasp.



(a) The depth map for the glass bowl without a contrast agent applied is noisy and leads to poor grasp hypothesis.

(b) When a contrast agent is applied, the depth map is clean and grasp performance improves. In our method, the contrast agent is only needed at scan time, not at pick time.

Fig. 7 Depth map for a transparent object with and without a contrast agent.

After the robot detects an initially successful grab, it shakes the object vigorously to ensure that it would not fall out during transport. After releasing the object and moving away, the robot checks to make sure the object is not stuck in its gripper. If the object falls out during shaking or does not release properly, the grasp is recorded as a failure. If the object is stuck, the robot pauses and requests assistance before proceeding.

Most objects have more than one pose in which they can stand upright on the table. If the robot knocks over an object, the model taken in the reference pose is no longer meaningful. Thus, during training, we monitored the object and returned it to the reference pose whenever the robot knocked it over. In the future, we aim to incorporate multiple components in the models which will allow the robot to cope with objects whose pose can change during training.

5.2.1 Contrast Agents

Many objects are challenging to grasp for our approach because they are transparent or dark in IR; grasping objects in spite of these issues is a challenging problem that remains an active area of research [11, 29]. As in Lysenkov et al. [24], which used paint to construct models of objects, apply a contrast agent to make the object visible in IR. Because our approach only uses depth information at scan time, and not at grasping time, and is heavily instance-based, it enables us to we apply a temporary coating for the IR scan and then remove it for learning visual models and later interaction with the object. (We found that hair spray coated with a layer of flour gives good performance, but can be easily removed.) Figure 7 shows an object and IR scan with and without a contrast agent. This demonstrates the advantage of our view-based and instance-based approach, which uses IR only during the single, initial scan, and not at inference time; once a good scan is obtained, grasps can be proposed at any time in the future and high-quality grasps can be learned through training. For the glass bowl in Figure 7, pick accuracy improved from $0/10$ without a contrast agent to $8/10$ with a contrast agent applied during the IR scan only and removed before the pick trials.

5.2.2 Bandit-Based Adaptation

We evaluate our bandit-based adaptation step by allowing the robot to try grasps on the object until it either halts or reaches a maximum of 50 grasp attempts. Our algorithm used an accept threshold of 0.7, reject confidence of 0.95 and epsilon of 0.2. These parameters result in a policy that rejects a grasp after one failed try, and accepts if the first three picks are successful. Different observations of success and failure will cause the algorithm to try the grasp more to determine the true probability of success.

We report the performance of the robot at picking using the learned height for servoing, but without grasp learning, then the number of trials used for grasp learning by our algorithm, and finally the performance at picking using the learned grasp location and orientation. These results appear in Table 1.

Low-Performing Objects

	Before Learning	During Learning	After Learning
Garlic Press	0/10	8/50	2/10
Helicopter	2/10	8/39	3/10
Gyro Bowl	0/10	5/15	3/10
Big Syringe	1/10	13/50	4/10
Sippy Cup	0/10	6/50	4/10
Clear Pitcher	4/10	3/4	4/10
Red Bucket	5/10	3/3	5/10
Wooden Spoon	7/10	3/3	7/10
Dragon	8/10	5/6	7/10
Triangle Block	0/10	3/13	7/10
Bottle Top	0/10	5/17	7/10
Ruler	6/10	5/12	7/10

High-Performing Objects

	Before Learning	During Learning	After Learning
Epipen	8/10	4/5	8/10
Icosahedron	7/10	7/21	8/10
Stamp	8/10	3/3	8/10
Blue Salt Shaker	6/10	5/10	8/10
Wooden Train	4/10	11/24	8/10
Packing Tape	9/10	3/3	9/10
Purple Marker	9/10	3/3	9/10
Round Salt Shaker	1/10	4/16	9/10
Toy Egg	8/10	4/5	9/10
Yellow Boat	9/10	5/6	9/10
Vanilla	5/10	4/5	9/10
Brush	10/10	3/3	10/10
Red Bowl	10/10	3/3	10/10
Shoe	10/10	3/3	10/10
Whiteout	10/10	3/3	10/10
Metal Pitcher	6/10	7/12	10/10
Mug	3/10	3/4	10/10
Syringe	9/10	6/9	10/10

Table 1 Results from the robotic evaluation of Prior Confidence Bound, sorted by pick success rate. All objects either maintained or improved performance after learning except for one: Dragon.

Some objects significantly improved performance. Objects that improved typically had some feature that prevented our grasping model from working. For example, the triangular block failed with the prior grasp because the gripper slid over the sloped edges and pinched the block out of its grippers. The robot tried grasps until it found one that targeted the sides that were parallel to the grippers, resulting in a flush grasp, significantly improving accuracy. For the round salt shaker, the robot first attempted to grab the round plastic dome, but the gripper is not wide enough for this grasp. It tried grasps until it found one on the handle that worked reliably.

Objects such as the round salt shaker and the bottle top are on the edge of tractability for thorough policies such as Thompson sampling. Prior Confidence Bound, on the other hand, rejects arms quickly so as to make these two objects train in relatively short order while bringing even more difficult objects such as the sippy cup and big syringe into the realm of possibility. It would have taken substantially more time and picks for Thompson sampling to reject the long list of bad grasps on the sippy cup before finding the good ones.

The garlic press is a geometrically simple object but quite heavy compared to the others. The robot found a few grasps which might have been good for a lighter object, but it frequently shook the press out of its grippers when confirming grasp quality. The big syringe has some good grasps which are detected well by the prior, but due to its poor contrast and transparent tip, orientation servoing was imprecise and the robot was unable to learn well due to poor signal. What improvement did occur was due to finding a grasp which consistently deformed the bulb into a grippable shape regardless of the perceived orientation of the syringe. We observed similar problems with the clear pitcher and icosahedron.

Objects that failed to improve fall into several categories. For some, performance was already high, so there was not much room to move or a reasonable grasp was accepted quickly without waiting to find a better one. A common failure mode for poorly performing objects was failure to accurately determine the position and orientation through visual servoing. If the grasp map cannot be localized accurately, significant noise is introduced because the map does not correspond to the same physical location on the object at each trial. For example, there is only about a 5mm difference between the width of the dragon and the width of the gripper; objects such as these would benefit from additional servo iterations to increase localization precision. If we double the number of iterations during fine grained servoing we can more reliably pick it, but this would either introduce another parameter in the system (iterations) or excessively slow down other objects which are more tolerant to error.

6 Related Work

Pick-and-place has been studied since the early days of robotics [2, 23]. Initial systems relied on models of object pose and end effector pose being provided to the algorithm, and simply planned a motion for the arm to grasp. Modern approaches use object recognition systems to estimate pose and object type, then libraries of grasps either annotated or learned from data [36, 8, 30]. These approaches attempt to create systems that can grasp arbitrary objects based on learned visual features or known 3D configuration.

Collecting training sets is an expensive process and is not accessible to the average user in a non-robotics setting. If the system does not work for the user's particular application, there is no easy way for it to adapt or relearn. Our approach enables the robot to autonomously acquire more information to increase robustness at detecting and manipulating the specific object that is important to the user at the current moment. Other approaches that focus on object discovery and manipulation fail to combine a camera that moves with an end to end system that learns to recognize objects and improves grasp success rates through experience [25, 16, 6, 37].

We formalize the problem as an N-armed bandit [41] where the robot aims to perform best arm identification [1, 5], or alternatively, to minimize simple regret after a finite exploration period [3]. Audibert and Bubeck [1] explored best arm identification in a fixed budget setting; however a fixed budget approach does not match our problem, because we would like the robot to stop sampling as soon as it has improved performance above a threshold. We take a fixed confidence approach as in Chen et al. [5], but their fixed confidence algorithm begins by pulling each arm once, a prohibitively expensive operation on our robot. Instead our algorithm estimates confidence that one arm is better than another, following Hoeffding races [28] but operating in a confidence threshold setting that incorporates prior information. By incorporating prior information, our approach achieves good performance without being required to pull all the arms. Kaufmann et al. [12] describe Bayesian upper confidence bounds for bandit problems but do not use simple regret, with a training period followed by an evaluation period. Additionally these approaches do not provide a stopping criterion, to decide when to move to the next object.

By formalizing grasp identification as a bandit problem, we are able to leverage existing strategies for inferring the best arm. Our system brings together key techniques in autonomous data collection and online learning for persistent robotic systems to establish a baseline grasping system which we show to be useful and extensible. Nguyen and Kemp [31] learn to manipulate objects such as a light switch or drawer with a similar self-training approach. Our work autonomously learns visual models to detect, pick, and place previously unencountered rigid objects by actively selecting the best grasp point with a bandit based system, rather than acquiring models for the manipulation of articulated objects. We rely on the fixed structure of objects rather than learning how to deal with structure that can change during manipulation.

Hudson et al. [10] used active perception to create a grasping system capable of carrying out a variety of complex tasks. Using feedback is critical for good performance, but the model cannot adapt itself to new objects. Existing general purpose grasp algorithms achieve fairly good performance on novel objects but leave appreciable gaps which could be closed by using our system to learn from experience [33, 34, 26, 9, 21]. Kroemer et al. [18] also use reinforcement learning to choose where to grasp novel objects, operating in continuous state spaces. However their approach does not incorporate prior knowledge and requires forty or more trials to learn a good grasp; in contrast, because our approach incorporates prior knowledge, we often obtain improvement after trying only a few grasps.

Collet et al. [6] describe an approach for lifelong robotic object discovery, which infers object candidates from the robot's perceptual data. This system does not learn grasping models and does not actively acquire more data to recognize, localize, and grasp the object with high reliability. It could be used as a first-pass to our system, after which the robot uses an active method to acquire additional data enabling it to grasp the object. Some approaches integrate SLAM and moving object tracking to estimate object poses over time but have not been extended to manipulation [42, 7, 35, 38]. Crowd-sourced and web robotics have created large databases of objects and grasps using human supervision on the web [14, 13]. These approaches outperform automatically inferred grasps but still require humans in the loop. Our approach can incorporate human annotations in the form of the prior: if the annotated grasps work well, then the robot will quickly converge and stop sampling; if they are poor grasps, our approach will find better ones.

7 Conclusion

The contribution of this paper is a system for automatically acquiring instance-based models of objects using a Baxter robot. Using our approach, the robot scans objects and collects RGB and depth information, which it then uses to perform detection, classification, and grasping. We demonstrate that the robot can improve grasping performance through active exploration by formalizing the grasping problem as best arm identification on an N-armed bandit. This approach significantly improves the robot's success rate at grasping specific objects as it practices picking them.

A limitation of our system is the requirement that a target object be in a canonical upright position with respect to the table, leaving only one degree of freedom to describe its orientation and two for its position. In our evaluation, if the salt shaker fell down, we reset it to a randomized upright position. With this paradigm, if we want to be able to handle the salt shaker whether it is upright or on its side, we must train two models and use logic outside the system to identify the models as the same object. Our next goal is to automatically explore these object modes and acquire classification, localization, grasping, and transition models for them over a long period of time. This improvement will enable any Baxter robot to automatically scan objects for long periods of time.

A second limitation is that by taking an instance-based approach, knowledge obtained from interacting with one object does not generalize to another object. Our approach runs on a stock Baxter robot and does not require any additional sensing. We aim to release our software so that anyone with a Baxter can train models using our approach and automatically share their models through a common database. This approach will enable us to scale up and distribute the scanning effort, so that a very large corpus of instance-based models can be automatically collected. As more and more models are collected, containing RGB image crops, point clouds, and logs of grasp success rates at different geometries, this data set will provide a unique opportunity to train new category-based models for general detection and grasping, supplying well-annotated data of multiple views of many instances of individual objects.

Our system focuses on manipulation of small objects; however, objects in the environment have more affordances than just manipulation: bottles can be opened; light switches can be flipped; buttons can be pushed, and doors can be unlocked. We aim to expand our approach to instance-based semantic mapping of large-scale environments, so that the robot can interactively learn about features of its environment such as drawers, door knobs, and light switches. By taking an instance-based approach, the robot can automatically create robust detectors for object features and fix up any problems through interaction with the environment. This approach will create a true semantic map of the environment, including affordances of objects.

References

[1] Jean-Yves Audibert and Sébastien Bubeck. Best arm identification in multi-armed bandits. In *COLT-23th Conference on Learning Theory-2010*, pages 13–p, 2010.
[2] Rodney A Brooks. Planning collision-free motions for pick-and-place operations. *The International Journal of Robotics Research*, 2(4):19–44, 1983.
[3] Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. Pure exploration in multi-armed bandits problems. In *Algorithmic Learning Theory*, pages 23–37. Springer, 2009.
[4] John Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):679–698, 1986.
[5] Shouyuan Chen, Tian Lin, Irwin King, Michael R Lyu, and Wei Chen. Combinatorial pure exploration of multi-armed bandits. In *Advances in Neural Information Processing Systems*, pages 379–387, 2014.
[6] Alvaro Collet, Bo Xiong, Corina Gurau, Martial Hebert, and Siddhartha S. Srinivasa. Herbdisc: Towards lifelong robotic object discovery. *The International Journal of Robotics Research*, 2014. doi: 10.1177/0278364914546030.
[7] Garratt Gallagher, Siddhartha S Srinivasa, J Andrew Bagnell, and Dave Ferguson. Gatmo: A generalized approach to tracking movable objects. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 2043–2048. IEEE, 2009.
[8] Corey Goldfeder, Matei Ciocarlie, Hao Dang, and Peter K Allen. The columbia grasp database. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 1710–1716. IEEE, 2009.
[9] Ilaria Gori, Ugo Pattacini, Vadim Tikhanoff, and Giorgio Metta. Three-finger precision grasp on incomplete 3d point clouds. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 5366–5373. IEEE, 2014.
[10] Nicolas Hudson, Thomas Howard, Jeremy Ma, Abhinandan Jain, Max Bajracharya, Steven Myint, Calvin Kuo, Larry Matthies, Paul Backes, Paul Hebert, et al. End-to-end dexterous manipulation with deliberate interactive estimation. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 2371–2378. IEEE, 2012.
[11] Ivo Ihrke, Kiriakos N Kutulakos, Hendrik PA Lensch, Marcus Magnor, and Wolfgang Heidrich. State of the art in transparent and specular object reconstruction. In *EUROGRAPHICS 2008 STAR–STATE OF THE ART REPORT*. Citeseer, 2008.
[12] Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On bayesian upper confidence bounds for bandit problems. In *International Conference on Artificial Intelligence and Statistics*, pages 592–600, 2012.
[13] David Kent and Sonia Chernova. Construction of an object manipulation database from grasp demonstrations. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 3347–3352. IEEE, 2014.
[14] David Kent, Morteza Behrooz, and Sonia Chernova. Crowdsourcing the construction of a 3d object recognition database for robotic grasping. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 4526–4531. IEEE, 2014.
[15] Object Recognition Kitchen. http://wg-perception.github.io/object_recognition_core/, 2014.
[16] Dirk Kraft, Renaud Detry, Nicolas Pugeault, Emre Baseski, Frank Guerin, Justus H Piater, and Norbert Kruger. Development of object and grasping knowledge by robot exploration. *Autonomous Mental Development, IEEE Transactions on*, 2(4):368–383, 2010.
[17] Michael Krainin, Peter Henry, Xiaofeng Ren, and Dieter Fox. Manipulator and object tracking for in-hand 3d object modeling. *The International Journal of Robotics Research*, 30(11):1311–1327, 2011.
[18] OB Kroemer, Renaud Detry, Justus Piater, and Jan Peters. Combining active learning and reactive control for robot grasping. *Robotics and Autonomous Systems*, 58(9):1105–1116, 2010.
[19] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1817–1824. IEEE, 2011.
[20] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. A scalable tree-based approach for joint object and pose recognition. In *Twenty-Fifth Conference on Artificial Intelligence (AAAI)*, August 2011.
[21] Quoc V Le, David Kamm, Arda F Kara, and Andrew Y Ng. Learning to grasp objects with multiple contact points. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 5062–5069. IEEE, 2010.
[22] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.
[23] Tomás Lozano-Pérez, Joseph L. Jones, Emmanuel Mazer, and Patrick A. O'Donnell. Task-level planning of pick-and-place robot motions. *IEEE Computer*, 22 (3):21–29, 1989.
[24] Ilya Lysenkov, Victor Eruhimov, and Gary Bradski. Recognition and pose estimation of rigid transparent objects with a kinect sensor. *Robotics*, page 273, 2013.
[25] Natalia Lyubova, David Filliat, and Serena Ivaldi. Improving object learning through manipulation and robot self-identification. In *Robotics and Biomimetics (ROBIO), 2013 IEEE International Conference on*, pages 1365–1370. IEEE, 2013.
[26] Alexis Maldonado, Ulrich Klank, and Michael Beetz. Robotic grasping of unmodeled objects using time-of-flight range data and finger torque information. In *IROS*, pages 2586–2591, 2010.
[27] Shie Mannor and John N Tsitsiklis. The sample complexity of exploration in the multi-armed bandit problem. *The Journal of Machine Learning Research*, 5: 623–648, 2004.
[28] Oded Maron and Andrew W Moore. Hoeffding races: Accelerating model selection search for classification and function approximation. *Robotics Institute*, page 263, 1993.
[29] Fabrice Mériaudeau, Rindra Rantoson, David Fofi, and Christophe Stolz. Review and comparison of non-conventional imaging systems for three-dimensional digitization of transparent objects. *Journal of Electronic Imaging*, 21(2):021105–1, 2012.
[30] Antonio Morales, Eris Chinellato, Andrew H Fagg, and Angel Pasqual del Pobil. Experimental prediction of the performance of grasp tasks from visual features. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 4, pages 3423–3428. IEEE, 2003.
[31] Hai Nguyen and Charles C Kemp. Autonomously learning to visually detect where manipulation will succeed. *Autonomous Robots*, 36(1-2):137–152, 2014.
[32] F. W. J. Olver, D. W. Lozier, R. F. Boisvert, and C. W. Clark, editors. *NIST Handbook of Mathematical Functions*. Cambridge University Press, New York, NY, 2010.
[33] Deepak Rao, Quoc V Le, Thanathorn Phoka, Morgan Quigley, Attawith Sudsang, and Andrew Y Ng. Grasping novel objects with depth segmentation. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 2578–2585. IEEE, 2010.
[34] Anis Sahbani and Sahar El-Khoury. A hybrid approach for grasping 3d objects. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 1272–1277. IEEE, 2009.
[35] Renato F Salas-Moreno, Richard A Newcombe, Hauke Strasdat, Paul HJ Kelly, and Andrew J Davison. Slam++: Simultaneous localisation and mapping at the level of objects. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1352–1359. IEEE, 2013.
[36] Ashutosh Saxena, Justin Driemeyer, and Andrew Y Ng. Robotic grasping of novel objects using vision. *The International Journal of Robotics Research*, 27(2): 157–173, 2008.
[37] David Schiebener, Julian Schill, and Tamim Asfour. Discovery, segmentation and reactive grasping of unknown objects. In *Humanoids*, pages 71–77, 2012.
[38] Antonio HP Selvatici and Anna HR Costa. Object-based visual slam: How object identity informs geometry. 2008.
[39] Irwin Sobel. An isotropic 3x3x3 volume gradient operator. Technical report, Technical report, Hewlett-Packard Laboratories, 1995.
[40] Richard Szeliski. *Computer vision: algorithms and applications*. Springer, 2010.
[41] William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, pages 285–294, 1933.
[42] Chieh-Chih Wang, Charles Thorpe, Sebastian Thrun, Martial Hebert, and Hugh Durrant-Whyte. Simultaneous localization, mapping and moving object tracking. *The International Journal of Robotics Research*, 26(9):889–916, 2007.