Bandit-Based System Adaptation for Instance-Based Grasping

Abstract

A key aim of current research is to create robots that can reliably manipulate generic 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. The contribution of this paper is to automate this adaptation process by formalizing a grasping system as a hierarchy of bandit problems, enabling the robot to apply learning techniques to adapt itself to manipulate specific objects with increased accuracy. The robot performs best arm identification using a variant of Hoeffding races, enabling it to quickly find an optimal arm and then move on to the next object. Our approach also incorporates knowledge from a prior defined in terms of an ordering on the arms. We demonstrate that our bandit-based adaptation step significantly improves accuracy over a non-adaptive system, enabling a robot to quickly and autonomously acquire models for objects, and adaptively improve them through experience.

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. Systems for general-purpose manipulation are computationally expensive and do not enjoy high accuracy on novel objects [37]. Instance-based approaches that focus on specific instances of objects can have higher accuracy but require training by a human operator, which is time consuming and can be difficult for a non-expert to perform [21, 25, 26]. Existing approaches to autonomously learn 3D object models still require expensive ICP-based methods to localize objects, which are susceptible to local minima and take time to converge [23].

To address this problem, we present an approach that enables a robot to learn to identify and grasp on a per object basis by adapting its perceptual framework to specific objects. Our grasping and percpetion pipeline uses standard computer vision techniques to perform data collection, feature extraction, and training, along with active visual servoing for localization. This framework succeeds to some degree with many objects, but reaches performance ceilings for specific objects for a variety of reasons. To address these limitations, we present a mathematical formalization of a system of software components combined with validators as an n-armed bandit problem [44]. Conventional abstractions and data flow used in system design translate into independence assumptions in the model; because we are operating using a robot which can actively collect additional information about the environment (although sometimes at higher cost in time and sensing), we can obtain a high-quality reward signal for supervision.

Unlike traditional bandit problems, where the agent trades off exploration and exploitation, in our formulation, the robot aims to minimize simple regret after a finite exploration period [7]. We follow Maron and Moore [30] to explore until the agent is confident that it has found an acceptable arm. Our approach enables the robot to quickly converge when it finds a good grasp, so it can move on to the next object.

Our evaluation demonstrates that a Baxter robot can autonomously learn robust models for detection and grasping, using its IR sensor and arm camera as a seven-degree of freedom one-pixel RGB-D camera. After training, Baxter can quickly and reliably grasp objects anywhere in its work space using closed-loop visual servoing in response to a person's requests. We demonstrate that our baseline system can learn to pick up objects approximately XX% of the time; augmenting this system with our self-training approach increases accuracy to YY%.

2 Grasping System

We first describe our instance-based object detection and pose estimation pipeline, which uses standard computer vision algorithms combined to achieve a simple software architecture, a high frame rate, and high accuracy at recognition and pose estimation. This pipeline can be manually trained by an expert to reliably detect and grasp objects. Section 2.6 describes our approach to enabling a robot to autonomously train this pipeline by actively collecting images and training data from the environment using Thompson sampling.

Our recognition pipeline takes video from the robot, proposes a small number of candidate object bounding boxes in

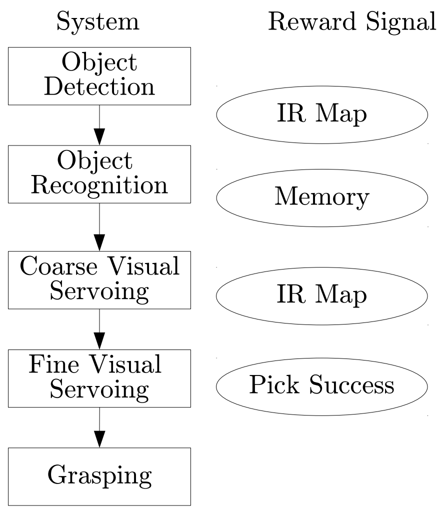


Figure 1: Data flow in our grasping system.

each frame, and classifies each candidate bounding box as belonging to a previously encountered object class. Our object classes consist of object instances rather than pure object categories. Using instance recognition means we cannot reliably detect categories, such as "mugs," but the system will be able to detect, localize, and grasp the specific instances for which it has models with much higher speed and accuracy. A visualization of data flow in the pipeline appears in Figure 1. **JGO: This doesn't quite seem current:** For each module, we formalize its input, output, and reward function; each component can have multiple implementations which better for different objects. The following sections describe how we can use this pipeline to learn which implementation to use for specific objects; this learning dramatically speeds up performance.

2.1 Object Detection

The input of the object detection component is an image, I; the output is a set of candidate bounding boxes, B. For object detection we use a modified Canny algorithm which terminates before the usual non-maximal suppression step [8].

We start by converting I to YCbCr opponent color representation. Then we apply 5×5 Sobel derivative filters [41] 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. After this we downsample, apply the two Canny thresholds, and find connected components. If a connected component is contained in another, we discared the contained component. We throw out boxes which do not contain enough visual data to classify. We generate a candidate bounding box for each remaining component by taking the smallest box which contains the component.

ST: Need to make a figure showing this process (analgous to the objectness figure)

2.2 Object Classification

The object recognition 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 [27] from all boxes of all classes and use k-means to extract a visual vocabulary of SIFT features [43]. 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 [43]. 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

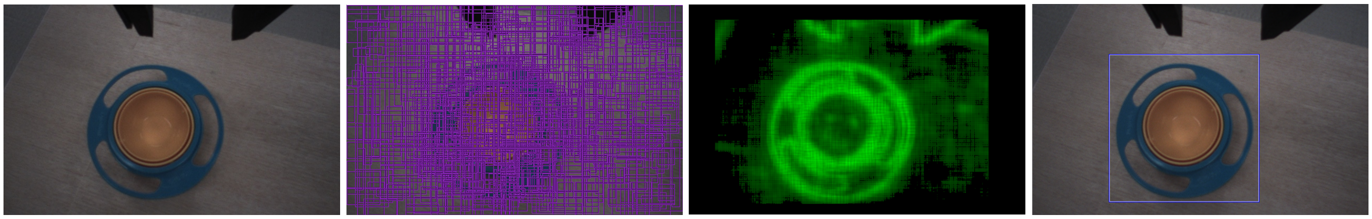
We use the image gradient for object detection and pose estimation. During object detection, the gradient of the whole image is the first step in the Canny pipeline. For pose estimation, we require a crop of image gradient of the object at a specific, known pose.

As during bounding box proposal, We approximate the gradient using 5×5 Sobel derivative filters [41], 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.

In order to match our template image with the crop observed at pick time, we remove the mean from and L^2 normalize the template and the crop. Removing the mean provides invariance to bias, and normalizing introduces invariance to scaling, which both help account somewhat for lighting.

2.4 Identifying Grasp Points

To identify a grasp points, we combine a depth map of the object with a model of the gripper. The depth map appears in



(a) Raw image from the camera. (b) Candidate bounding boxes from Bing. (c) Integral image objectness map. (d) Candidate bounding boxes.

Figure 2: The object detection pipeline, showing a raw image from the camera, the integral image computed using objectness, and candidate bounding boxes.

Figure **??** and is acquired by moving the rangefinder on the arm through a raster scan over the object. The grasp model scores each potential grasp according to a linear model of the gripper 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 assume only crane grasps rather than full 3d grasping, where θ is the angle which the gripper assumes for the grasp.

2.5 Closed-Loop Grasping

To grasp an object, we first scan the work area by moving the camera until the object is detected an recognized. Then we perform active visual servoing to move the arm directly above the object. Next, we perform orientation servoing using the pose estimation algorithm. Because these components are instance-based, they report position and orientation with high accuracy, enabling us to use a proportional controller (with no derivative or integral terms) to move the arm into position.

2.6 Autonomous Training

An object model in our framework consists of the following elements:

- cropped object templates (roughly 200), $t^1 ... t^K$
- depth map, D, which consists of a point cloud, (x, y, z, r, g, b).
- cropped gradient template, t_0

Additionally, the model can be augmented with words or attributes, $w_1 ... w_n$ which people might use to describe the object, so the robot can respond to natural language commands such as "Put the cup on the left table."

To train our model, the robot first moves the object to a known pose, then acquires images that are annotated with a pose as well as a cropped bounding box for training. As typical in machine learning applications, the more images we can acquire, from the more viewpoints, the more accurate our detection, pose estimation, and grasping. To achieve this accuracy, the robot autonomously collects this information by using a depth sensor to acquire an initial grasp, move the object to a standard known pose, and then actively collect data for view-based methods. Our learning algorithm appears in Algorithm **??**.

Once the object has been moved to a known pose, we acquire the object model by moving the camera to a sequence of prespecified orientations. We automatically crop the image using our object detection system.

3 Bandit-based Adaptation

Our formal model of a system defines a reinforcment learning problem, in which the action space consists of different settings for system parameters, and the reward function consists of the output of the testing modules for each component. The modular design of our system supports a decomposition in the RL problem, so that we can treat each optimization problem as a separate N-armed bandit problem. In many systems, the problem with this formulation is obtaining a reward function to test different parameter values: the behavior of the system must be measured by a person, and if an autonomous approach existed to produce accurate output, it would obviate the need for the system in the first place. In contrast, in the robotic domain, many edges in the system diagram can be verified, at the cost of time or additional sensing. This technique enables us to obtain high-quality supervision at relevant edges in the system diagram, supporting the decomposition into a bandit problem. For example, optimizing grasp height at level one is rewarded based on its accuracy at identifying bounding boxes for the object; this training step does not require running the entire pipeline. Thus our algorithm for adaptation runs from the root to the leaves of the system diagram, optimizing each parameter based on the tester closest to it in the system tree.

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 $>= k$) 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 an ordering on the arms, promising ones first. This order corresponds to a prior, but the agent is only required to specific that promising arms are first, not assign any specific probability distribution on μ.

Our algorithm is to iterate through each arm, and try it until we have identified it either is above k with probability c (in which case it terminates) or it is below k with probability c (in which case it moves to the next arm in the sequence. To compute this probability we need to estimate the probability that the true payout probability, μ is greater than the threshold, c given the observed number of successes and failures:

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

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

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

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 [33].

```
BestArmConfidence(armOrder, δ, k)
Initialize S_0 ... S_n to 0
Initialize F_0 ... F_n to 0
for i ∈ armOrder do
    r ← sample(arm_i)
    if r = 1 then
        S_i ← S_i + 1
    end
    else
        F_i ← F_i + 1
    end
    p_reject ← ∫_0^k Pr(μ_i = u|S_i, F_i)dμ
    p_accept ← 1 − p_reject
    if p_reject ≥ k then
        break ;              // go to the next arm
    else if p_accept ≥ k then
        return i ;           // accept this arm
    else
        pass ;               // keep pulling this arm
    end
end
```

Algorithm 1: Algorithm for Best Arm Identification

Note that if $\mu = k$ the runtime is unbounded, so we fix an $\epsilon > 0$ and accept a grasp if $\mu \in [k - \epsilon, k + \epsilon]$ with probability c.

4 Robotic Implementation

We use stock Baxter and one computer. We use the wrist cameras to obtain RGB at 30hz. Different parts of the visual pipeline run at different frequencies, but the full visual system runs at 2Hz in typical conditions. To obtain the 3D point cloud maps, we use the IR sensor in conjunction with the RGB measurement. Color triangulation, a nice image of the brush. Quickly describe raster scan. Parzen kernel density estimator, downsampling to fill in data.

Figure: A nice scan of the brush together with an RGB image of the brush from a similar height.

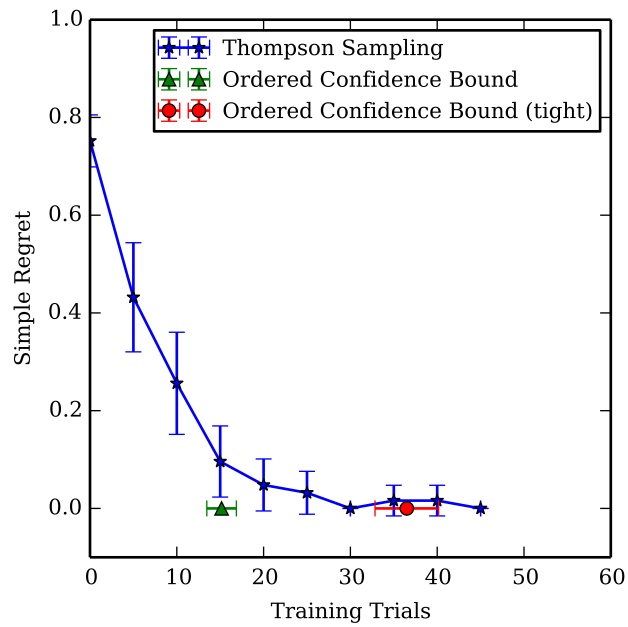


Figure 3: Results comparing our approach to various baselines in simulation.

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 first assess our approach in simulation, comparing it to Thompson sampling and a fixed policy. Next we describe our robotic evaluation and manipulation which assesses our system's ability to learn and adaptively improve its ability to grasp objects, end-to-end.

5.1 Simulation

Our simulated results compare our approach to Thompson sampling as well as a fixed policy, assessing the tradeoffs inherent in our choice of parameters and algorithm. We present results in Table 3. We simulate picking performance by creating a sequence of 20 bandits, where each arm pays out at a rate of 0.1 except for one, which pays out at 0.9. We move the location of the best arm to be either second, half-way through the sequence, or last. These correspond to a highly informative prior, an uninformative prior, and a pathological prior where the best arm is last.

5.2 Robotic Evaluation

We have implemented our approach on the Baxter robot, which is equipped with a seven-degree-of-freedom arm with a camera and IR depth sensor, which we use as a one-pixel depth camera to acquire our models.

The robot acquired visual and RGB-D models for N objects using our autonomous learning system. 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

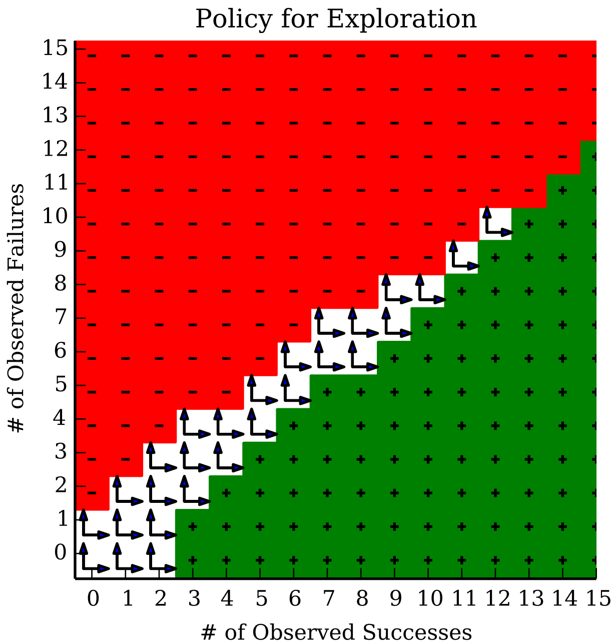


Figure 4: Policy for the agent given a belief state, defined by an observed number of suceseses and failures on that arm for our model parameters. Red (-) shows states where the agent rejects the arm and move to the next one; green (+) shows states where it accepts the arm and stops learning, and the arrows show movement to the next belief state based on whether the next trial is successful.

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 Algortihm 1.

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.

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. The policy for exploring an arm appears in Figure **??**.

6 Related Work

Bohg et al. [4] survey data-driven approaches to grasping. Our approaches can be thought of as a pipeline for automatically building an experience database consisting of object models and known good grasps, using analytic approaches to grasping unknown objects to generate a grasp hypothesis space and bandit-based methods for trying grasps and learning instance-based distributions for the grasp experience database. In this way our system achieves the best of both approaches: models for grasping unknown objects can be applied; when they do fail, the system can automatically recover by trying grasps and adapting itself based on that specific object. Additionally, we can use other grasp detectors to seed our prior, since it is only based on an ordering and does not require an explicit probability distribution. We can also interface with manual labeling similar to the forklift demarcation: circle an area and we will restrict exploration to that region.

	Prior	Training	Marginal
Epipen	8/10	4/5	8/10
Toy Egg	8/10	4/5	9/10
Vanilla	5/10	4/5	9/10
Triangle block	0/10	3/13	7/10
Metal pitcher	6/10	7/12	10/10
Helicopter	2/10	8/39	3/10
Salt shaker	1/10	4/16	9/10
Clear pitcher	4/10	3/4	4/10
Gyro bowl	0/10	5/15	3/10
Sippy cup	0/10	6/50	4/10
Total	34/100	48/164	66/100
Rate	0.34	0.29	0.66

Table 1: Results from the robotic evaluation.

Ude et al. [45] described an approach for detecting and manipulating objects to learn models. It uses a bag of words model and learns to detect the objects. It does not learn a model for grasping. Schiebener et al. [39] describes an extension that also does model learning. The robot pushes the object and then trains an object recognition system. It does not use a camera that moves and does not grasp. Schiebener et al. [38] discovers and grasps unknown objects.

Summary:

- People doing SLAM. Wang et al. [47], Gallagher et al. [13],
- People doing 3d reconstruction. Krainin et al. [23], Banta et al. [2]
- People doing big databases for category recognition. Kent et al. [20], Kent and Chernova [19], Lai et al. [26], Goldfeder et al. [14]
- Object tracking in vision (typically surveillance).
- POMDPs for grasping. Platt et al. [34], Hsiao et al. [16]
- People doing systems. Hudson et al. [17], Ciocarlie et al. [11]

Crowd-sourced and web robotics have created large databases of objects and grasps using human supervision on the web [20, 19]. These approaches outperform automatically inferred grasps but still require humans in the loop. Our approach enables a robot to acquire a model fully autonomously, once the object has been placed on the table.

Zhu et al. [48] created a system for detecting objects and estimating pose from single images of cluttered objects. They use KinectFusion to construct 3d object models from depth measurements with a turn-table rather than automatically acquiring models.

Chang et al. [9] created a system for picking out objects from a pile for sorting and arranging but did not learn object models.

next-best view planning [24]

Nguyen and Kemp [32] learn to manipulate objects such as a light switch or drawer with a similar self-training approach. Our work learns visual models for objects for autonomous pick-and-place rather than to manipulate objects.

Developmental/cognitive robotics [29**?**]

Banta et al. [2] constructs a prototype 3d model from a minimum number of range images of the object. It terminates reconstruction when it reaches a minimum threshold of accuracy. It uses methods based on the occluded regions of the reconstructed surfice to decide where to place the camera and evaluates based on the reconstruction rather than pick up success. Krainin et al. [23] present an approach for autonomous object modeling using a depth camera observing the robot's hand as it moves the object. This system provides a 3d construction of the object autonomously. Our approach uses vision-based features and evaluates based on grasp success. Eye-in-hand laser sensor. [**?**]

ST: Need to find the instance-based work that Erik mentioned when he said it was a "solved problem."

Velez et al. [46] created a mobile robot that explores the environment and actively plans paths to acquire views of objects such as doors. However it uses a fixed model of the object being detected rather than updating its model based on the data it has acquired from the environment.

Methods for planning in information space [15, 1, 35] have been applied to enable mobile robots to plan trajectories that avoid failures due to inability to accurately estimate positions. Our approach is focused instead on object detection and manipulation, actively acquiring data for use later in localizing and picking up objects. **ST: May need to say more here depending on what GRATA actually is.**

Early models for pick-and-place rely on has been studied since the early days of robotics [6, 28]. These 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 [37, 14, 31]. These approaches attempt to create systems that can grasp arbitrary objects based on learned visual features or known 3d configuration. Collecting these 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, instead, 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.

Visual-servoing based methods [10] **ST: Need a whole paragraph about that.**

ST: Ciocarlie et al. [11] seems highly relevant, could not read from the train's wifi. Existing work has collected large database of object models for pose estimation, typically curated by an expert [25]. Kasper et al. [18] created a semiautomatic system that fuses 2d and 3d data, but the setup requires a special rig including a turntable and a pair of cameras. Our approach requires an active camera mounted on a robot arm, but no additional equipment, so that a robot in the home can autonomoulsy acquire new models.

?] describes 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 reliabilitiy. It could be used as a first-pass to our system, after which the robot uses an active method to acquire additional data enablign it to grasp the object. Approaches that integrate SLAM and moving object tracking estimate pose of objects over time but have not been extended to manipulation [47, 13, 36, 40].

Our approach is similar to the philosphy adopted by Rethink Robotic's Baxter robot, and indeed, we use Baxter as our test platform [12]. **ST: Haven't actually read this paper, just making stuff up based on Rod's talks. Should read the paper and confirm.** Baxter's manufacturing platform is designed to be easily learned and trained by workers on the factory floor. The difference between this system and our approach is we rely on the robot to autonomously collect the training information it needs to grasp the object, rather than requiring this training information to be provided by the user.

Robot systems for cooking [5, 3] or furniture assembly [22] use many simplifying assumptions, including pre-trained object locations or using VICON to solve the perceptual system. We envision vision or RGB-D based sensors mounted on the robot, so that a person can train a robot to recognize and manipulate objects wherever the robot finds itself.

Approaches to plan grasps under pose uncertainty [42] or collect information from tacticle sensors [16] using POMDPs. **?**] describe new algorithms for solving POMDPs by tracking belief state with a high-fidelity particle filter, but using a lower-fidelity representation of belief for planning, and tracking the KL divergence.

Hudson et al. [17] 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.

7 Conclusion

ST: First paragraph: contributions. What are the things this paper has done to advance the state of the art?

ST: Next paragraphs: future work, spiraling upward to more and more ambitiuos extensions.

Right now, NODE runs on Baxter. We will port NODE to PR2 and other AH systems. GRATA could be applied in other domains as well. What are some examples?

Ideas for doing for the paper, otherwise future work:

- Semantic mapping.
- Detection and manipulation in clutter and occlusion.
- Amazon mechanical turk for labels, so we can follow commands and gesture.

- Object tracking over time so we can answer questions about what happened to the object.
- Object oriented SLAM so we can handle joint localization and mapping.
- Semantic mapping of objects over time. Deciding when to go look again, maintaining history, etc.
- Scaling to lots and lots of objects.
- Using the database of lots and lots of objects to do category recognition.
- Multiple poses during training (e.g., what happens when you drop the object?)

References

[1] Nikolay Atanasov, Jerome Le Ny, Kostas Daniilidis, and George J Pappas. Information acquisition with sensing robots: Algorithms and error bounds. *arXiv preprint arXiv:1309.5390*, 2013.

[2] Joseph E Banta, LR Wong, Christophe Dumont, and Mongi A Abidi. A next-best-view system for autonomous 3-d object reconstruction. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 30(5):589–598, 2000.

[3] Michael Beetz, Ulrich Klank, Ingo Kresse, Alexis Maldonado, L Mosenlechner, Dejan Pangercic, Thomas Ruhr, and Moritz Tenorth. Robotic roommates making pancakes. In *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, pages 529–536. IEEE, 2011.

[4] Jeannette Bohg, Antonio Morales, Tamim Asfour, and Danica Kragic. Data-driven grasp synthesis—a survey. 2013.

[5] Mario Bollini, Stefanie Tellex, Tyler Thompson, Nicholas Roy, and Daniela Rus. Interpreting and executing recipes with a cooking robot. In *Proceedings of International Symposium on Experimental Robotics (ISER)*, 2012.

[6] Rodney A Brooks. Planning collision-free motions for pick-and-place operations. *The International Journal of Robotics Research*, 2(4):19–44, 1983.

[7] Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. Pure exploration in multi-armed bandits problems. In *Algorithmic Learning Theory*, pages 23–37. Springer, 2009.

[8] John Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):679–698, 1986.

[9] Lillian Chang, Joshua R Smith, and Dieter Fox. Interactive singulation of objects from a pile. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3875–3882. IEEE, 2012.

[10] François Chaumette and Seth Hutchinson. Visual servo control. i. basic approaches. *Robotics & Automation Magazine, IEEE*, 13(4):82–90, 2006.

[11] Matei Ciocarlie, Kaijen Hsiao, Edward Gil Jones, Sachin Chitta, Radu Bogdan Rusu, and Ioan A Şucan. Towards reliable grasping and manipulation in household environments. In *Experimental Robotics*, pages 241–252. Springer, 2014.

[12] C Fitzgerald. Developing baxter. In *Technologies for Practical Robot Applications (TePRA), 2013 IEEE International Conference on*, pages 1–6. IEEE, 2013.

[13] 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.

[14] 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.

[15] Ruijie He, Sam Prentice, and Nicholas Roy. Planning in information space for a quadrotor helicopter in a gps-denied environment. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 1814–1820. IEEE, 2008.

[16] Kaijen Hsiao, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Task-driven tactile exploration. Robotics: Science and Systems Conference, 2010.

[17] 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.

[18] Alexander Kasper, Zhixing Xue, and Rüdiger Dillmann. The kit object models database: An object model database for object recognition, localization and manipulation in service robotics. *The International Journal of Robotics Research*, 31(8):927–934, 2012.

[19] 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.

[20] 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.

[21] Object Recognition Kitchen. http://wg-perception.github.io/object_recognition_core/, 2014.

[22] Ross A. Knepper, Stefanie Tellex, Adrian Li, Nicholas Roy, and Daniela Rus. Single assembly robot in search of human partner: Versatile grounded language generation. In *Proceedings of the HRI 2013 Workshop on Collaborative Manipulation*, 2013.

[23] 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.

[24] Simon Kriegel, T Bodenmliller, Michael Suppa, and Gerd Hirzinger. A surface-based next-best-view approach for automated 3d model completion of unknown objects. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 4869–4874. IEEE, 2011.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] Oded Maron and Andrew W Moore. Hoeffding races: Accelerating model selection search for classification and function approximation. *Robotics Institute*, page 263, 1993.

[31] 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.

[32] Hai Nguyen and Charles C Kemp. Autonomously learning to visually detect where manipulation will succeed. *Autonomous Robots*, 36(1-2):137–152, 2014.

[33] 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. Print companion to [**?**].

[34] Robert Platt, Leslie Kaelbling, Tomas Lozano-Perez, and Russ Tedrake. Simultaneous localization and grasping as a belief space control problem. In *International Symposium on Robotics Research*, volume 2, 2011.

[35] Samuel Prentice and Nicholas Roy. The belief roadmap: Efficient planning in belief space by factoring the covariance. *The International Journal of Robotics Research*, 2009.

[36] 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.

[37] 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.

[38] David Schiebener, Julian Schill, and Tamim Asfour. Discovery, segmentation and reactive grasping of unknown objects. In *Humanoids*, pages 71–77, 2012.

[39] David Schiebener, Jun Morimoto, Tamim Asfour, and Aleš Ude. Integrating visual perception and manipulation for autonomous learning of object representations. *Adaptive Behavior*, 21(5): 328–345, 2013.

[40] Antonio HP Selvatici and Anna HR Costa. Object-based visual slam: How object identity informs geometry. 2008.

[41] Irwin Sobel. An isotropic 3x3x3 volume gradient operator. Technical report, Technical report, Hewlett-Packard Laboratories, 1995.

[42] Freek Stulp, Evangelos Theodorou, Jonas Buchli, and Stefan Schaal. Learning to grasp under uncertainty. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 5703–5708. IEEE, 2011.

[43] Richard Szeliski. *Computer vision: algorithms and applications*. Springer, 2010.

[44] 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.

[45] Ales Ude, David Schiebener, Norikazu Sugimoto, and Jun Morimoto. Integrating surface-based hypotheses and manipulation for autonomous segmentation and learning of object representations. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1709–1715. IEEE, 2012.

[46] Javier Velez, Garrett Hemann, Albert S Huang, Ingmar Posner, and Nicholas Roy. Active exploration for robust object detection. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, page 2752, 2011.

[47] 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.

[48] M. Zhu, N. Atanasov, G. Pappas, and K. Daniilidis. Active Deformable Part Models Inference. In *European Conference on Computer Vision (ECCV)*, 2014.