Autonomously Acquiring Instance-Based Object Models

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 contribution of this paper is a system that automates this data collection and adaptation process. When the robot encounters a novel object, it automatically collects instance based models for detecting, estimating pose, and grasping that object. This approach achieves the generality of category-based methods with the reliability of instance-based methods, at the cost of time spent collecting data for the model. We demonstrate that our approach enables an unmodified Baxter robot to autonomously acquire models for a wide variety of objects and then robustly respond to pick-and-place requests for those objects.

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 [35]. 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 [20, 24, 25]. Existing approaches to au-

Brown University

tonomously learn 3D object models still require expensive ICP-based methods to localize objects, which are susceptible to local minima and take time to converge [22].

To address these problems, we present an approach that captures the high accuracy of instance-based methods without the need for manually acquiring training data by enabling a robot to learn to identify and grasp on a per object basis. Our grasping and perception pipeline uses standard computer vision techniques to perform data collection, feature extraction, and training, along with active visual servoing for localization. Using our algorithm, the robot detects candidate objects for training using a depth sensor, then actively collects view-based visual templates to perform robust instance-based object detection, pose estimation and grasping models using visual servoing. Because our camera can move with seven degrees of freedom, the robot can collect large quantities of data leading to simple visual models that perform with high accuracy. Our approach is enabled by three components: our end-to-end algorithm for collecting view-based training data with supervision obtained from a higher-reliability depth sensor, which is supported by a simple and robust method for determining candidate grasps using a depth sensor mounted on a seven-degree-of-freedom arm, along with an approach for autonomously and reliably finding object bounding boxes once the object is on a background such as a floor or table.

2 Overview and Approach

Robots can interact with the physical world and collect multiple images for processing. We exploit this fact to make some computer vision tasks easier by physically adding invariance rather than hand engineering features.

Our approach lets us use rich but inconvenient depth sensors once at learning time and then exploit the knowledge they provide by using only an RGB camera at run time.

2.1 Invariance in Images

A theme in computer vision and pattern recognition Is the notion of invariance. We like to detect objects anywhere in an image, invariant to their positions. We want to find the spoon no matter which direction it is pointing, invariant to its pose. Shadows and reflections can be confusing so we want to be invariant to lighting. What if you could add invariance to an image? With a robot you can.

Canny Autotune is robotic because we have access to the environment and can take additional images.

We can actively light the area with white or IR light (Kinect, for instance) to get some invariance to lighting. The IR rangefinder we use is active.

Subsystem	Invariance(s) Addressed
Canny Servo	Position
Gradient Servo	Orientation
Light Map	Lighting Gain, Bias, Reflection, Shadow

SIFT features give some invariance to scale, as their name implies. BoW models give invariance to small model deformation. For invariance to large deformations, significant and expensive modeling such as DPM or CNN is required.

We can use our freedom as roboticists (all 7 degrees of it) to add invariance to a task not by addressing the features or models as much prior work does, but by adding it to the data at training and inference.

Collecting the data in a timely fashion both requires and facilitates a relatively tight loop of control over the robots joint states. Canny servoing will have an auto-threshold feature that adds lighting invariances to a fast saliency map. The resulting movement yields invariance to large translations. Gradient servoing provides invariance to orientation and small translations.

Light Map is what averaging before the gradient servo will become. We map the projection of the object onto the table in physical coordinates, aggressively ignoring reflections and shadows, moving to get a better view and fill in gaps. Then we use that map for inference, our immediate cases being that we treat it as an image and classify it then feed it into gradient servo.

Deterministic light mapping is a nice non-parametric method, but you could imagine an online method that iteratively estimates the orientation of the object, uses prior knowledge of the appearance combined with accumulated measurements to construct the light map, then uses that map to estimate the orientation of the object, and repeats until convergence. Such a method might be called Light SLAM.

2.2 Sensors

By developing systems that work well in real time in simple environments with specific objects, we obtain the capacity to collect enough labeled data to train systems with the ability to generalize and perform well in more complicated, cluttered environments. This approach also lets us get a head start with the logistical problems involved in transforming theory to practice.

Infrared (IR) imaging can be useful for segmenting and determining geometry and many devices (including the range finder we employ) are active in that they shine IR light and measure the return signal. The rangefinders readings are somewhat invariant to color and material while motionless, but show anisotropic artifacts near edges and during movement. None the less, for the most part the readings are good enough for us to infer grasps.

Transparent materials are invisible, reflective materials induce artifacts, and dark materials are worse during movement. To overcome these issues on other devices, people have spraypainted objects to make them permanently well behaved. This works to a degree but really breaks the suspension of disbelief and is inappropriate for domestic application. We developed a temporary contrast agent that can be applied safely and easily removed. We scan objects in IR once during training and use that scan during inference so that we can pick things up using only RGB data and do not require a permanent agent.

We use a two phase contrast agent consisting of a binding phase and a conditioning phase. The conditioning phase scatters and reflects incoming IR light and the binding phase attach the conditioner to the surface of the object. For binders we investigated vaseline, zinc oxide ointment (diaper creme), and Aquanet hairspray. The only conditioner we tried was wheat flour, but you could substitute cornstarch or baking powder in a pinch. We settled on the Aquanet because it was fast, not too messy, and as the shell of flour hardened it eventually shrank so much that it detached cleanly on its own from many surfaces after 60 minutes. Aquanet may be inappropriate for some surfaces, in which case water could serve as a binder. In extreme situations, since the scan is top down and only needs to detect normal (horizontal) surfaces, you can run a scan with no binding phase since a dusting of powder will settle on desirable surfaces. This could even be seen as physical feature detection.

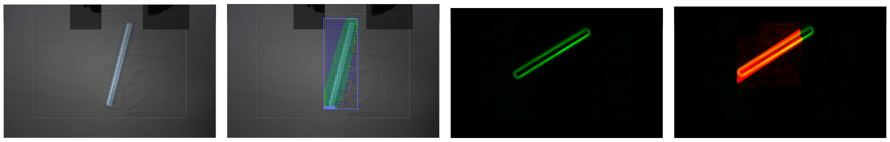
2.3 Purpose

JGO: I know some of this doesn't really belong but I had these thoughts and wanted to add them in case they are useful for, say, future work. One purpose of this system is to investigate an approach to gathering data and training models that enable multiple robotic platforms to robustly detect and manipulate novel objects and to improve in ability over time. The approach consists of three phases.

In the first phase, the system trains only instance level models of objects and collects data on those objects as it manipulates them, using heuristic proposals for grasp points and confidence bounds for learning. In the second phase, the system uses the data collected from instance level models to train category level detectors for object class and grasp locations. During the second phase, the system still uses its parent proposals during operation but it continuously evaluates the performance of the juvenile category level detectors. When the category level detectors are comparable to the parent detectors, the third phase begins and the system uses its trained category models instead of the originally provided parent heuristics.

In addition to being a convenient way to initially train models, this three fold process is a general method for bootstrapping a new classifier to replace an old one without taking the system offline.

3 Grasping System



(a) RGB Image of an object. (b) Bounding box extracted for the object. (c) Gradient image of the object. (d) Stored gradient image overlayed on the object image after servoing.

Fig. 1 Results at each phase of the localization pipeline.

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. Section 3.6 describes our approach to enabling a robot to autonomously collect the data needed to perform grasping with this pipeline.

Our recognition pipeline takes video from the robot, 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. 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 2 while Figure 1 shows results from each phase of the localization pipeline. 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.

3.1 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 [7]. We start by converting I

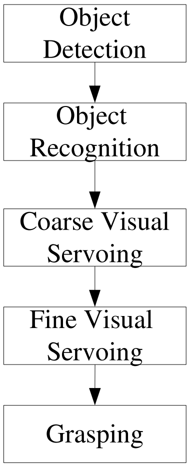


Fig. 2 Data flow in our grasping system.**ST: maybe combine with figure 1**

to 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. 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.

3.2 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 [26] from all boxes of all classes and use k-means to extract a visual vocabulary of SIFT features [41]. 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 [41]. 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.

3.3 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.

We denote the gradient by

$$\Delta I = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right) \tag{1}$$

As in bounding box proposal, 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 3.1 and 3.2.

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.

3.4 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 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.

3.5 Closed-Loop Grasping

To grasp an object, we first scan the work area by moving the camera until the object is detected and 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. Last, we move the arm to the desired grasp location, close the gripper, and pick up the object.

3.6 Autonomous Training

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

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

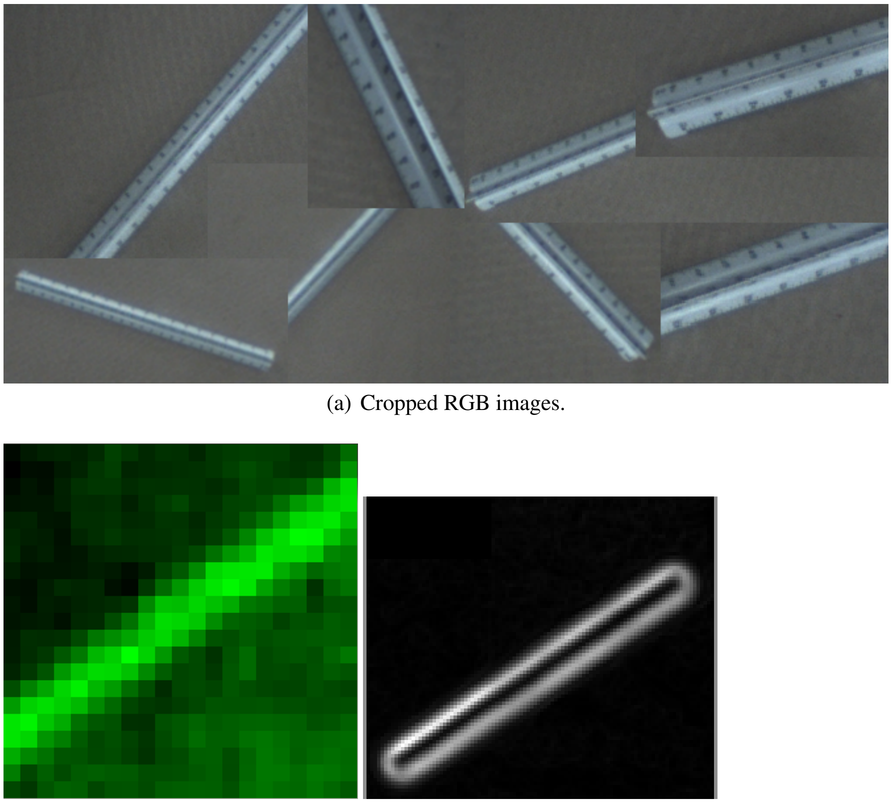
The robot collects gradient images by servoing to the center of the extracted bounding box for the object, described in Section 3.1, and then recording a gradient image at several different heights. It records each image for several frames to average away noise from the camera. Gradient images for the 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, turing the arm into a seven degree of freedom, one-pixel depth sensor. We perform an IR scan 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,

Once the object has been moved to a known pose, we acquire the object model by moving the camera around the object, extracting bounding boxes from the resulting imgages, and storing the resulting crops. Figure 3(a) shows RGB images automaticaly collected for one object in our dataset.

4 Evaluation

We evaluate our approach in two ways, first at mapping a tabletop scene, and second by assessing picking accuracy. Video showing our training and grasping pipeline is available at https://www.youtube.com/watch?v=xfH0B3g782Y.



(a) Cropped RGB images.

(b) Depth map.

(c) Gradient map.

Fig. 3 An object model in our pipeline.

4.1 Mapping Accuracy

Mapping assesses the ability of our robot to accurately localize and label objects in a tabletop scene. We assess performance by creating scenes containing a subset of the objects used in our evaluation. The robot maps the scene by maintaining a data structure 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 category to the map. Figure 4.1 shows the map created in this way for a tabletop scene.

We report precision and recall for each object in the scene. Precision means that if the system associated that object with a bounding box, it correctly labeled it. Recall means that the object associated each object in the scene with a bounding box with a correct label.

(a) Tabletop scene. (b) Map created from the scene.

4.2 Picking Accuracy

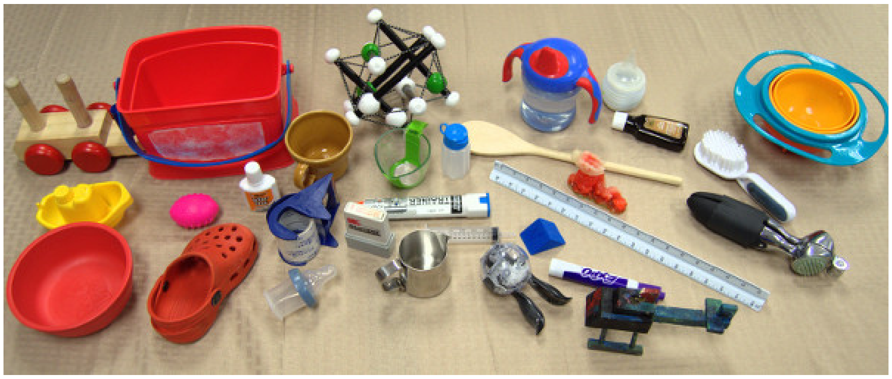


Fig. 4 The objects used in our evaluation.

We evaluated our system by training models for a diverse set of objects and assessing picking accuracy. The objects in our evaluation appear in Figure 4.

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. We report the performance of the robot at picking after training models for the system.

Our evaluation demonstrates that for many objects, the system obtains a very high grasp success rate. Objects failed for several reasons. Some objects, such as the garlic press, were quite heavy, at the edge of the robot's capability to lift. Other objects, such as those in Figure 5 are poorly visible in IR. In our other work [31], we showed that we could improve the robot's grasp rate on these objects by actively

Object	# Picks/# Tries
Brush	10/10
Red Bowl	10/10
Shoe	10/10
Whiteout	10/10
Yellow Boat	9/10
Syringe	9/10
Packing Tape	9/10
Purple Marker	9/10
Stamp	8/10
Toy Egg	8/10
Dragon	8/10
Epipen	8/10
Icosahedron	7/10
Wooden Spoon	7/10
Blue Salt Shaker	6/10
Metal Pitcher	6/10
Ruler	6/10
Vanilla	5/10
Red Bucket	5/10
Wooden Train	4/10
Clear Pitcher	4/10
Mug	3/10
Helicopter	2/10
Round Salt Shaker	1/10
Big Syringe	1/10
Bottle Top	0/10
Garlic Press	0/10
Gyro Bowl	0/10
Sippy Cup	0/10
Triangle Block	0/10

Table 1 Results from the robotic evaluation. We tested on 30 objects. Overall performance is 165 picks of 300 tries, or 55%.

picking and learning better grasp points. This demonstrates the advantage of our vision-based approach, which uses IR only at grasp proposal time, and not at inference time; once a good grasp is obtained, high-quality grasps can be inferred at pick time.

Besides our learning approach, we also found that applying a contrast agent significantly improves grasp success. For example, for the two objects in Figure 5, performance using a contrast agent raises to XX.

Object	# Picks/# Tries (no contrast agent)	# Picks/# Tries (contrast agent)
Round Salt Shaker	1/10	
Bottle Top	0/10	
Overall		

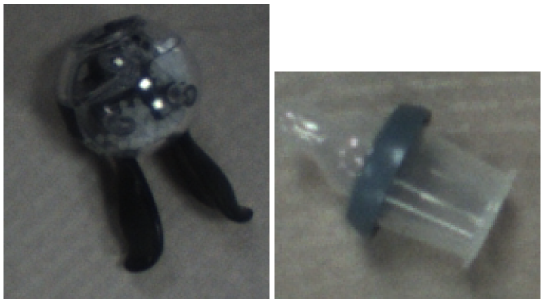


Fig. 5 Poorly performing objects because they are not well visible in IR.

5 Related Work

JGO: I can do a little survey here on standard approaches for adding invariance in computer vision and attempt to find some approaches in robotics that add invariance in atypical ways.

Bohg et al. [4] survey data-driven approaches to grasping. Our approach 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 using our bandit-based method 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 attempt to recover by trying grasps and adapting itself based on that specific object.

Ude et al. [42] 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. [37] 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. [36] discovers and grasps unknown objects.

Summary:

- People doing SLAM. Wang et al. [44], Gallagher et al. [12],
- People doing 3d reconstruction. Krainin et al. [22], Banta et al. [2]
- People doing big databases for category recognition. Kent et al. [19], Kent and Chernova [18], Lai et al. [25], Goldfeder et al. [13]
- Object tracking in vision (typically surveillance).
- POMDPs for grasping. Platt et al. [32], Hsiao et al. [15]
- People doing systems. Hudson et al. [16], Ciocarlie et al. [10]

Crowd-sourced and web robotics have created large databases of objects and grasps using human supervision on the web [19, 18]. 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. [45] 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. [8] created a system for picking out objects from a pile for sorting and arranging but did not learn object models.

next-best view planning [23]

Nguyen and Kemp [30] 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 [28**?**]

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 surface to decide where to place the camera and evaluates based on the reconstruction rather than pick up success. Krainin et al. [22] 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. [43] 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 [14, 1, 33] 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, 27]. 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 [35, 13, 29]. 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 [9] **ST: Need a whole paragraph about that.**

ST: Ciocarlie et al. [10] 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 [24]. Kasper et al. [17] 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 autonomously 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 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. Approaches that integrate SLAM and moving object tracking estimate pose of objects over time but have not been extended to manipulation [44, 12, 34, 38].

Our approach is similar to the philosophy adopted by Rethink Robotic's Baxter robot, and indeed, we use Baxter as our test platform [11]. **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 [21] 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 [40] or collect information from tacticle sensors [15] 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. [16] 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.

6 Conclusion

The contribution of this paper is a system for automatically acquiring instance-based models of objects. We have demonstrated our system's performance at creating a map of objects in its local environment as well as by assessing the pick accuracy of objects on a challenging test set. Our approach runs on a stock Baxter robot and does not require any additional sensing, and we plan to release the source code once the paper is accepted, enabling anyone with a Baxter to train models using our approach.

We plan to extend our framework so that object models are automatically uploaded to a common database. 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, training on data of multiple views of many instances of individual objects.

A signficiant limitation of our existing system is the requirement for an object to always be in a consistent position with respect to the table. In our pick-and-place evaluation, we reset the object if it fell down (for example, if a salt shake fell on its side). Our next goal is to automatically detect these object modes and acquire detection, localization, and grasping models for them as well using active exploration.

Next, the robot can learn to transition an object from one mode to another, so that it can learn to robustly manipulate the objects. Using this framework, the system will be capable of autonomously learning grasping models over a long period of time, significantly increasing the size of the dataset collected and thus robustness.

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

[8] 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.

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

[10] 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.

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

[12] 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.

[13] 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.

[14] 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.

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

[16] 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.

[17] 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.

[18] 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.

[19] 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.

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

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

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

[31] John Oberlin and Stefanie Tellex. Bandit-based adaptation for robotic grasping. In *IJCAI (Under Review)*, 2015.

[32] 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.

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

[34] 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.

[35] 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.

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

[37] 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.

[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] 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.

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

[42] 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.

[43] 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.

[44] 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.

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