Autonomously Acquiring Models of Objects for Instance-Based Manipulation

Author Names Omitted for Anonymous Review. Paper-ID [add your ID here]

***Abstract*—Manipulating and reasoning about objects is an important task for robots that help people in the home, in factories, and in hospitals. A key aim of current research is to create robots that can reliably manipulate generic objects in clutter. 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. To address this problem, we focus not on *category-based* manipulation (pick up any mug) but rather *instance-based* manipulation (pick up this mug). Instance-based recognition and pose estimation using vision can be highly accurate but requires adapting the system to the specific object being manipulated: the system designer must adapt the system to specific objects in the environment in large and small ways, ranging from collecting training data to choosing parameter values to selecting algorithms for tasks such as image segmentation or bounding box classification. The contribution of this paper is to formalize this adaptation process as a hierarchy of bandit problems, enabling the robot to apply learning techniques such as Thompson sampling to learn to manipulate specific objects. Our framework runs on an unmodified Baxter robot; using our algorithm, a robot can interact with an object for twenty minutes, and then reliably and quickly localize it with vision and pick it up with closed-loop visual servoing. We demonstrate that the adaptation step significantly improves accuracy over a non-adaptive system.**

I. 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 [36]. 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 [22, 26, 27]. 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 [24].

To address this problem robustly, 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, training, along with active visual servoing for localization. This framework works 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 [42]. Conventional abstractions and data flow used in system design translate into independence assumptions in the model, enabling RL algorithms such as Thompson sampling to be applied to autonomously adapt the system for specific objects. Our system uses Thompson sampling to learn to adapt itself for specific objects, greatly improving end-to-end performance at picking after training is complete. We use slower, more accurate sensing approaches to provide supervision for faster, simpler methods that excel with large amounts of training data. For example, to perform grasping we use an analytic model to select grasp points, but depending on the object, the best grasp according to the analytic model may not be optimal; the robot can learn better grasps for that object using the analytic model as a prior and actively collecting data for that 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 50% of the time; augmenting this system with our self-training approach increases accuracy to XX%.

Our software is compatible with ROS and the Baxter SDK version 1.0.0, and we intend to release it as free software should our paper be accepted. As more and more instance-based models are collected, this corpus will form a unique training set for category-based models for detecting and grasping novel objects, since the robot will have a very large number of views of a large set of objects as well as storing depth information and grasping success.

II. 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 II-F 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.

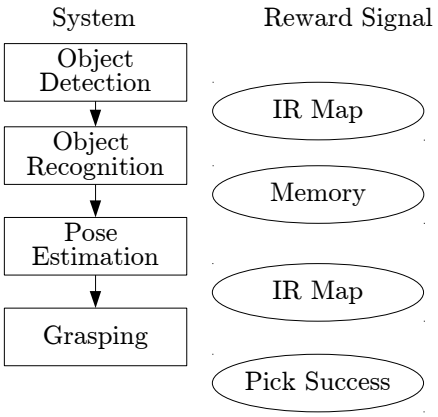


Fig. 1. Data flow in our grasping system.

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 1. 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.

A. Object Detection

The input of the object detection component is an image, I; the output is a set of candidate bounding boxes, B. We validate this component during training using the 3d IR map of the object, which we back-project into the original image to obtain the true bounding box. We use two algorithms for object detection, one based on the BING objectness metric, which works well for objects that have distinct colors, and one based on gradients, which works well for objects that contrast well with the background.

1) Detecting Objects Using BING and Integral Images: To detect candidate objects, we first apply the BING objectness detector [10] to the image, which returns a set $\{B_i\}$ of thousands of approximate object bounding boxes in the image, shown in Figure 2(b). This process substantially reduces the number of bounding boxes we need to consider but is still too large to process in real time. Besides, even good bounding boxes from BING are typically not aligned to the degree that we require. Therefore, we use integral images to efficiently compute the per-pixel map:

$$J(p) = \sum_{B \in \{B_i\} s.t. p \in B} \frac{1}{Area(B)}. \quad (1)$$

This map appears in Figure 2(c). We then apply the Canny edge detector with hysteresis [7] to find the connected components of bright regions in the map $J(p)$, which correspond with high probability to objects in the image. We form our candidate object bounding boxes by taking the smallest bounding box which surrounds each connected component, shown in Figure 2(d). These bounding boxes make it easy to gather training data and to perform inference in real time, but at the expense of poorly handing occulusion as overlapping objects are fused into the same bounding box. It is possible to search within the proposed bounding boxes to better handle occlusion. Figure 2 shows images from each step in this pipeline, ending with just one candidate bounding boxes for an objects on an empty table. Note that we designed this pipeline to quickly and accurately provide bounding boxes in the presence of relatively unobstructed backgrounds to support the training process; Section **??** describes our approach to recognition under clutter and occlusion.

2) Detecting Objects Using Image Gradients: **ST: John – please fill this part in.**

B. Object Recognition

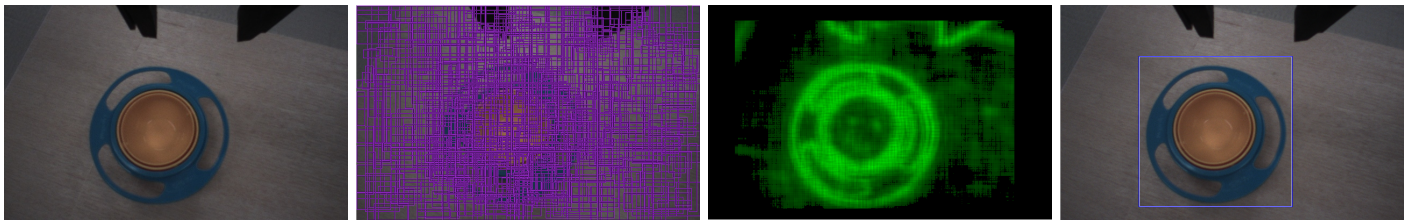
The object recognition module takes as input a bonding box, B, and outputs a label for that object, c, based on the robot's memory. This label is used to identify the object and look up other information about the object for grasping further down the pipeline.

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 [28] from all boxes of all classes and use k-means to extract a visual vocabulary of SIFT features [41]. We then construct a BoW 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. Existing approaches for instance-based grasping such as LINE-2D require the order of 2000 whereas our SIFT-based approach performs well with only 200 examples [16].

C. Pose Estimation

To perform pose estimation, we require an image gradient of the object at a specific, known pose:

$$\Delta I = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right) \quad (2)$$



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

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

We approximate the gradiant using differences after smoothing the training image:

$$\frac{\partial I(x,y)}{\partial x} \approx \frac{I(x+1,y) - I(x-1,y)}{2} \quad (3)$$

$$\frac{\partial I(x,y)}{\partial y} \approx \frac{I(x,y+1) - I(x,y-1)}{2} \quad (4)$$

To estimate pose, we perform data augmentation by rotating our training image and finding the closest match to the image currently recorded from the camera, as detected and localized via the pipeline in Section II-A and II-B.

D. 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 **??**. 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, 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); at this point we assume only crane grasps rather than full 3d grasping.

E. 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 servoign 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

F. 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, d).
- 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."

GraspObject()

while true **do**
  $I \leftarrow loadImage()$
  $B \leftarrow Attemptagrasp.$
  if grasp is successful **then**
    Move object to training area.
    Map(object).
  end if
end while

Fig. 3. The high-level object-learning algorithm.

Map

for $(x^k, y^k, z^k) \in scan$ **do**
  $I^k \leftarrow ImageAt(x^k, y^k, z^k)$
  $t^k \leftarrow Crop(I^k)$ using the approach described in Section II-A
  $D^k \leftarrow PointAt(x^k, y^k, z^k)$
end for

Fig. 4. The high-level algorithm for acquiring visual and grasping models of objects.

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 3.

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 integral images computed over the bounding boxes inferred by the Bing objectness detector.

ThompsonSample(module, tester)

for $action \in parameterSettings(module)$ **do**
  $S_{action} \leftarrow 0$
  $F_{action} \leftarrow 0$
end for
for $t \in 1, 2, ..., do$ **do**
  for $action \in parameterSettings(module)$ **do**
    $\mu_{action}(t) \leftarrow \text{Sample}(Beta(S_{action} + 1, F_{action} + 1))$
  end for
  $bestAction \leftarrow \underset{action}{\text{argmax}}\ \mu_{action}(t)$
  $r_t \leftarrow tester(bestAction)$
  if x **then**
    $S_{i(t)} \leftarrow S_{i(t)} + 1$
  end if
end for

Fig. 5. The high-level algorithm for acquiring visual and grasping models of objects.

III. 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 (which is true of most well-design systems) supports a decomposition in the RL problem, so that we can treat each optimization problem as a separate N-armed 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.

IV. Experimental Setup

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 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.

A. Object Detection and Pose Estimation

For object Detection and pose estimation, we constructed data sets on which we could evaluate our models. This involved hand annotating the ground truth for the images in the sets, which is a costly procedure which we are attempting to eliminate for future tasks. However, we cannot evaluate our system in a principled fashion without such a data set.

We demonstrate our method's success in this setting where we pay the cost to acquire the data so that we can trust our method and that cost need not be payed during future applications.

TABLE I
Performance of our system on the object detection task.

Data Collection Method	Success Rate
Expert Annotation	0.0
Dense Sampling	0.0
Hard Negatives Auto-Stopping	0.0

TABLE II
Performance of our system on offline data.

Data Set	
Expert Curated	0.0
Expert (noisy)	0.0
Automatic (curated)	0.0
Automatic (noisy)	0.0

Probably uses confusion rates as the objective function.
Expert viewpoint collection (uses at least hard negatives)
Super dense sampling
uniform hard negative sampling with stopping criterion

V. Evaluation and Discussion

We could report the performance of the system as a function of user interactions. We could report the performance of the system as a function of program lifetime. Our representative set could consist of a block, a spoon, a bowl, a diaper, and a sippy cup. A *single cut video* showing multiple grasps of all objects is available here.

A. Object Detection

We establish a baseline for performance by training the system in a representative domain specific setting, which tells us how well it can perform on laboratory objects when trained by an expert. This represents the best that the system could be expected to perform.

B. Pose Estimation

C. Grasping

D. PID Control

VI. 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.

Grasp Sampling Method	Success Rate
Expert Annotation	0.0
Uniform Sampling	0.0
Thompson Sampling	0.0

Fig. 6. Performance of our system on the grasping task.

Parameter Learning Method	Average Convergence Time
Expert Annotation	0.0
Constant Learning Rate	0.0
Decaying Learning Rate	0.0
Wide Scale Random Noise	0.0

Fig. 7. Performance of our system on the PID control task.

In this way our system achieves the best of both approaches: models for grasping unknown objects can be applied; when they do not fail, the system can automatically recover by trying grasps and adapting itself based on that specific object.

Ude et al. [43] 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 grapsing. Schiebener et al. [38] describes an extension that also does model learning. The robot pushes the object and then trains an object recognition system. It does nto use a camera that move and does not grasp. Schiebener et al. [37] discovers and grasps unknown objects.

Summary:

- People doing SLAM. Wang et al. [45], Gallagher et al. [13],
- People doing 3d reconstruction. Krainin et al. [24], Banta et al. [2]
- People doing big databases for category recognition. Kent et al. [21], Kent and Chernova [20], Lai et al. [27], Goldfeder et al. [14]
- Object tracking in vision (typically surveillance).
- POMDPs for grasping. Platt et al. [33], Hsiao et al. [17]
- People doing systems. Hudson et al. [18], Ciocarlie et al. [11]

Crowd-sourced and web robotics have created large databases of objects and grasps using human supervision on the web [21, 20]. 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. [46] 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 [25]

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 [30**?**]

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. [24] 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. [44] 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, 34] 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, 29]. 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 [36, 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 [9] **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 [26]. Kasper et al. [19] 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 [45, 13, 35, 39].

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 [23] 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 [17] 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. [18] 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.

VII. 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] 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] Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and Philip Torr. Bing: Binarized normed gradients for objectness estimation at 300fps. In *IEEE CVPR*, 2014.

[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] Stefan Hinterstoisser, Cedric Cagniart, Slobodan Ilic, Peter Sturm, Nassir Navab, Pascal Fua, and Vincent Lepetit. Gradient response maps for real-time detection of textureless objects. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(5):876–888, 2012.

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

[18] 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.

[19] 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.

[20] 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.

[21] 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.

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

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[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] 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.

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

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

[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] 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.

[43] 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.

[44] 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.

[45] 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.

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