Autonomously Acquiring Models of Objects for Instance-Based Manipulation and Mapping

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. General-purpose object manipulation requires object recognition, pose estimation, and grasp planning; existing solutions cannot reliably recognize or pick up an object the robot has never encountered before, while existing category-based computer vision approaches run much slower than real time and encounter many false positives and negatives. However in many applications, general-purpose object detection or manipulation is not required: the robot would be useful if it could adapt itself to recognize, localize, and manipulate the small set of objects most important in that application, but do so with very high reliability. To address this problem, we focus not on *category recognition* (pick up any mug) but rather *instance recognition* (pick up this mug). Instance recognition can be highly accurate but requires collecting training examples of each object we would like the robot to manipulate, which can be expensive, tedious, and error-prone. The contribution of this paper is an algorithm that enables a robot to autonomously collect training data for instance-based detection, localization, and active visual servoing for grasping. Using our algorithm, a robot can interact with an object for ten minutes, and then reliably and quickly localize it with vision and pick it up with closed-loop visual servoing. We demonstrate our learned model can enable efficient closed-loop grasping with high reliability, follow natural language pick-and-place commands, and enable vision-based semantic mapping using learned detectors.**

I. Introduction

Robotic assistants will assist us at childcare, help us cook, and provide service to doctors, nurses, and patients in hospitals. Many tasks require a robot to robustly perceive and manipulate objects. Some systems require training by a human operator on an object to object basis, which is time consuming and can be difficult for a non-expert to perform [12, 14, 15]. Systems which do not require training on a per object basis are computationally expensive and do not enjoy the highest accuracy compared to instance-based approaches and have not been demonstrated for grasping [19]. Existing approaches to learn 3d object models still require expensive ICP-based methods to localize objects, which are susceptible to local minima and take time to converge [**?**].

View-based methods for instance detection have many advantages over model-based methods, because they directly capture the visual appearance of the object, and are relatively simple and efficient to implement because they operate on low-level features [**?**]. However these systems require large amounts of training data for robust performance, for example more than 2000 images which must be manually collected and annotated with bounding boxes for the state of the art LINE2D method [**?**]. We address this problem by enabling a robot to learn to identify and grasp on a per object basis by autonomously collecting large amounts of supervised training data for instance-based visual detection and pose estimation. Our contribution is an algorithm which allows a robot to autonomously train its subsystems by using slower, more expensive sensing approaches to provide supervision for faster, simpler methods that excel with large amounts of training data.

To address these issues, we present an approach for enabling a robot to train its own view-based model to recognize and manipulate the specific objects it will need to use during collaborations with humans. 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 closed-loop grasping 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 even under occlusion. Our approach is enabled by three innovations: 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.

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 models can be used by a mobile robot for object detection and semantic mapping in cluttered environments.

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. Our software includes the capability to upload instance-based models to the RoboBrain database [**?**]; 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. Object Detection and Pose Estimation

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 III describes our approach to enabling a robot to autonomously train this pipeline by actively collecting images and training data from the environment.

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. Object Detection

To detect candidate objects, we first apply the BING objectness detector [6] to the image, which returns a set $\{B_i\}$ of thousands of approximate object bounding boxes in the image, shown in Figure 1(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:

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

This map appears in Figure 1(c). We then apply the Canny edge detector with hysteresis [?] 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 1(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 1 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 IV describes our approach to recognition under clutter and occlusion.

B. Object Recognition

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

C. Pose Estimation

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

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

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

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

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

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 **??**.

E. Closed-Loop Grasping

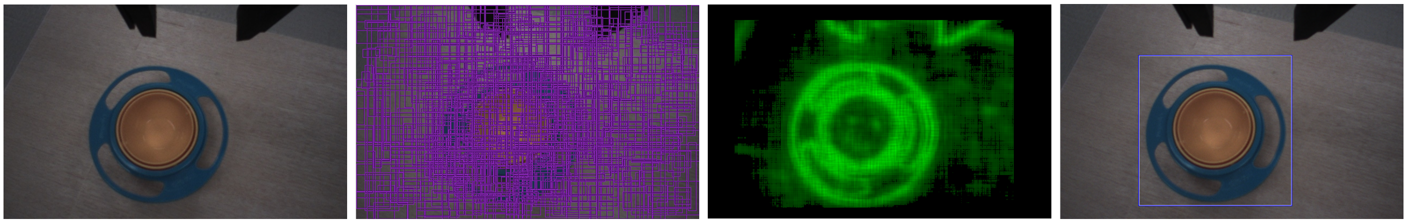
To grasp an object, we first scan the work area by moving the camera until the object is detected an recognized. Then we perform active visual servoign to move the arm directly above the object. Next, we perform orientation servoing using the pose estimation algorithm. Because these components are instance-based, 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

III. 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



(a) Raw image from the camera.

(b) Candidate bounding boxes from Bing.

(c) Integral image objectness map.

(d) Candidate bounding boxes.

Fig. 1. The object detection pipeline, showing a raw image from the camera, the integral image computed using objectness, and candidate bounding boxes.

```
Find Candidate Object
  while true do
    IR scan for grasp points.
    Attempt a grasp.
    if grasp is successful then
      Move object to training area.
      Map(object).
    end if
  end while
```

Fig. 2. 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. 3. The high-level algorithm for acquiring visual and grasping models of objects.

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 must first move the object to a known pose, then acquire 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 2.

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.

IV. Object Detection for Semantic Mapping

V. 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.

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

VI. 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.

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

B. Pose Estimation

C. Grasping

D. PID Control

VII. RELATED WORK

?] 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. **?**] 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. **?**] discovers and grasps unknown objects.

Summary:

- People doing SLAM. **? ?**],
- People doing 3d reconstruction. **? ?**]
- People doing big databases for category recognition. **? ?** Lai et al. [15], Goldfeder et al. [9]
- Object tracking in vision (typically surveillance).
- POMDPs for grasping. **? ?**]
- People doing systems. **?** Ciocarlie et al. [7]

Crowd-sourced and web robotics have created large databases of objects and grasps using human supervision on the web [**? ?**]. 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.

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

Fig. 4. 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. 5. Performance of our system on the PID control task.

?] created a system for detecting objects and estimating pose from single images of cluttered objects. They use Kinect-Fusion to construct 3d object models from depth measurements with a turn-table rather than automatically acquiring models.

?] created a system for picking out objects from a pile for sorting and arranging but did not learn object models.

next-best view planning [**?**]

?] 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 [**? ?**]

?] 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. **?**] 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. [20] 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 [11, 1, 18] 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 [4, 16]. 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 [19, 9, 17]. 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 [5] **ST: Need a whole paragraph about that.**

ST: Ciocarlie et al. [7] 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 [14]. **?**] 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 [**? ? ? ?**].

Our approach is similar to the philosphy adopted by Rethink Robotic's Baxter robot, and indeed, we use Baxter as our test platform [8]. **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 [3, 2] or furniture assembly [13] 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 [**?**] or collect information from tacticle sensors [**?**] 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.

?] 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.

VIII. 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] 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.

[3] 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.

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

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

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

[7] 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.

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

[9] 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.

[10] Sergio Guadarrama, Erik Rodner, Kate Saenko, Ning Zhang, Ryan Farrell, Jeff Donahue, and Trevor Darrell. Open-vocabulary object retrieval. In *Proceedings of Robotics: Science and Systems*, Berkeley, USA, July 2014.

[11] 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.

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

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

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

[19] 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.

[20] 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.