Autonomously Acquiring Models of Objects for Instance-Based Grasping

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

***Abstract*—Manipulating objects is an important task for robots that help people in the home, in factories, and in hospitals. General-purpose pick-and-place requires object recognition, pose estimation, and grasp planning; existing solutions cannot reliably recognize or pick up an object the robot has never encountered before []. However in many applications, general-purpose pick-and-place is not required: the robot would be useful if it could recognize and manipulate the small set of objects most important in that application, but do so with high reliability. To address this problem, we define a SLAM-based approach which enables the robot to actively acquire visual and grasp models of novel objects by moving its sensor and trying grasps, integrating information over time. Unlike conventional SLAM, in our model the hidden state is the object pose, and the map consists of invariant features of the object such as appearance information and grasp points. Our approach converts the task of *category recognition* (pick up any mug) to *instance recognition* (pick up this mug), enabling models to be autonomously acquired for specific objects. Using our approach, a robot can interact with an object for ten minutes, and then reliably localize (90%) and manipulate it (90% successful grasps).**

I. Introduction

Robotic assistants will assist us at childcare, help us cook, and provide service to doctors, nurses, and patients in hospitals. Many such tasks require a robot to robustly perceive and manipulate objects. Conventional systems are capable of perception and manipulation in a limited sense. 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 [20, 23, 24]. There do exist some systems which do not require training on a per object basis, but they are computationally expensive and do not enjoy the highest accuracy or precision and have not been demonstrated for grasping [13].

To obtain the benefits of both approaches, we propose a system which trains itself to recognize and manipulate the specific objects it will need to use during future collaborations with humans. Our system is powerful because it learns to identify and grasp on a per object basis. Our system is portable, convenient, and general because the expert knowledge it employs is built into the algorithms which it uses to train itself, requiring only basic interaction from a non-technical human collaborator.

Our contribution is an algorithm which allows a robot to autonomously train its subsystems, together with three applications of the algorithm to the tasks mentioned above. The first application is recognizing the category of an object and the second application is estimating the pose of the object, both of which we accomplish with simple and robust computer vision algorithms combined with our proposed algorithm for autonomous collection of training data. The third application is grasping the object, which we accomplish with visual servoing techniques. Each of these components is well understood in its own right, and existing methods allow expert users to train systems to accomplish these tasks satisfactorily.

When we apply the algorithm to the recognition task, the robot trains the recognition system to discriminate between object instance categories. When we apply the algorithm to the pose estimation task, the robot trains the pose estimation system to determine which pose an identified object holds. When we apply the algorithm to the grasping task, the robot trains the grasping system to successfully and quickly pick and place the target object. Crucially, our algorithm can recognize when it is doing a poor job at learning and asks a human collaborator to manually annotate information in those cases.

It works because our experiments tell us so. This is how well they work: . Thus we see an improvement over expert trained systems (cite usability results), and is an improvement over un-annotated systems (cite success rates, the ability to generalize, and the low computational overhead since we don't crunch expensive features or do heavy classification at run time).

ST: We have to say the right things about ORK. Why does ORK suck? Why doesn't it already solve the problem?

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. Additionally, section **??** describes our approach to enabling a robot to autonomously train this pipeline by actively collecting images and training data from the environment.

A. Object Recognition

JGO: Cover BING, SIFT, BoW, typical training pipelines, and RGB-D approaches popular in the robotics community.

Our recognition pipeline takes RGB-D 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.

To generate candidate bounding boxes, we first apply the BING objectness detector [7] to the image, which returns a set $\{B_i\}$ of thousands of approximate object bounding boxes in the image. This process substantially reduces the number of bounding boxes we need to consider but is still too large for us 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:

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

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. 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.

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 [].

The use of SIFT features is motivated by the instance level nature of our task. State-of-the-art vision methods typically use HOG [] or CNN [] features, but that choice is motivated by category level recognition. **ST: What about category level recognition motivates HOG or CNN? Can you be more specific?**

We use kNN because it is easy to rebuild online, which is a key property a classifier should enjoy if it is to interact with our framework in real time. State-of-the-art computer vision classifiers currently employ SVM's [] or other models which require expensive training. Using such a model would introduce a training step in the inside loop of our data collection process, which would be costly in either engineering or time. It is possible to use kNN during the online collection process and then train a stronger classifier in the background at higher latency, essentially introducing a cascading step in the data collection process.

B. Pose Estimation

JGO: Computer vision approaches, geometry and point cloud based approaches. Dieter Fox's automatic training pipeline (how well developed is it? we may need to sell our approach as being a general algorithm for allowing a robot to train arbitrary subsystems in order to differentiate ourselves.) We tackle the pose estimation problem using the same classification pipeline that we use for object recognition. We train a separate pose classifier for each object class. This time, the class assigned to each training example is the orientation from which the object is viewed in that example. During inference, we first determine the object class of a candidate bounding box, and once the class is known we apply the corresponding pose classifier to determine the orientation from which we are viewing the object. We combine this orientation with position information from the point cloud information derived from the D channel of the RGB-D video to form a full pose estimate.

C. Object Grasping

JGO: Including open and closed loop paradigms, learning specific and generic grasp models.

We consider a setting where a robotic arm with 7 degrees of freedom grasps objects with a parallel plate gripper which adds an additional degree of freedom, but much of what we discuss could be extended to other arms and grippers, for instance the universal jamming gripper [].

We use a dual rate PID controller in the sense that we use two sets of PID coefficients. The first set is for making large adjustments when the aim if off by a significant amount. The second set is for making small adjustments when aim is close to the target.

Our system is distributed and thus at times there is an appreciable amount of latency between communicating components. Care is taken to synchronize robot movement with object detection reports, allowing only a fixed amount of movement per report.

Visual servoing tutorial paper: [6]

ST: Cite the visual servoing tutorial paper and talk about the connection to grasping rectangles.

This Grasp Rectangle business fits in nicely with the reticle.

0. Estimate the depth of the table by inspecting non-object locations. This helps decide when to close the gripper.

1. Servo orientation to the '0' orientation, or one of a few sparsely sampled keypoint orientations. Each viewing orientation (from the wrist) is tied to a grasping orientation.

2. Servo to a 'normal' scale. This is fixed, we don't want multiple scales running around.

3. Now instead of aiming at the center, aim at a proposed target offset from the center.

D. PID Controller

JGO: Maybe a coordinate descent algorithm or wide-scale random noise search. Since we use a dual-rate controller, there are two separate sets of coefficients that we must train. We train the high-rate coefficients with the objective of getting the aim within the "close" threshold. We train the low-rate coefficients with the objective of getting the aim

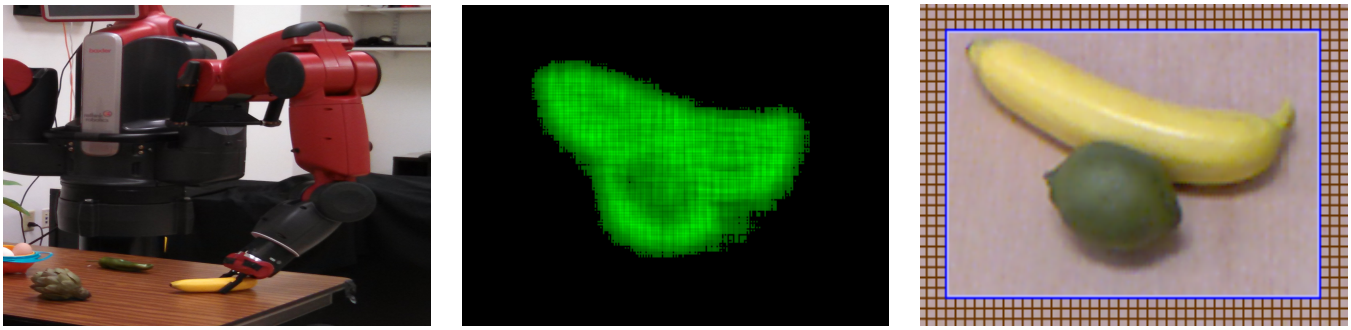

Fig. 1. The Object Detection Pipeline. Left: The raw image as viewed through the kinect. Center: The computed objectness map J. Right: Labeled object detections. **ST: Use subfigure**

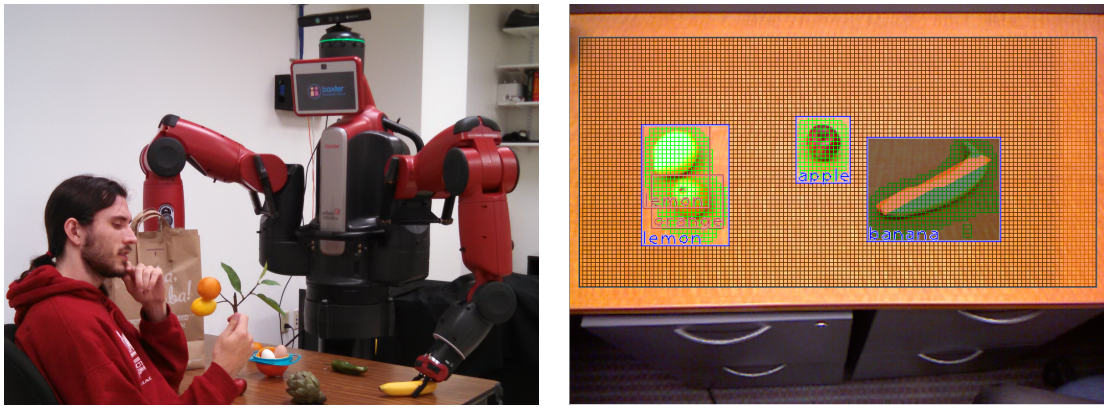

Fig. 2. Left: Baxter uses visual servoing to grasp an object. Right: The view through Baxter's wrist camera, showing the location of the target as well as the objective reticle.

within the 'hit' threshold. The training for the high and low-rate coefficients is analogous and happens independently, so without loss of generality we describe the training process for an arbitrary set of coefficients.

JGO: I imagine this has been done before so it would be good to find who did this. ST: Find someone to learn PID coefficients. A single set of PID coefficients consists of K_P, K_I, K_D. In the inside loop of EM-like training, we randomly pick a coefficient K to train, fix the other two coefficients, and use a local search algorithm [] to find the optimal value of K conditioned on the fixed values of the other two parameters. This problem is not necessarily convex and so we run the inside loop of our algorithm until we have converged to a local minimum.

III. Simultaneous Localization and Mapping of Tabletop Objects

We aim to autonomously acquire instance-based models of objects based on exploration. To carry out this inference, we follow approaches based on recursive state estimation [34] and SLAM [9]. In our approach, the map that the robot is acquiring is not an occupancy grid or landmark locations, as in conventional SLAM, but rather knowledge of the appearance and grasp points of the objects being mapped. The localization problem is estimating the pose of these objects at each time step. We can also jointly estimate the pose of the robot's end effector, but this pose is often known with high accuracy in the local coordinate system. The graphical model for our approach appears in Figure 3.

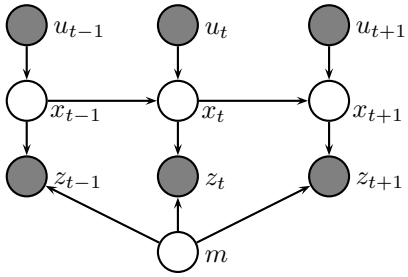

Fig. 3. Graphical model for our proposed approach.

We define the following variables:

$x_t = \langle r_t, o_1 \dots o_k \rangle$. Our state consists of the pose of the robot's end effector, r_t, along with the pose and category of all the objects currently being tracked, $o_1 \dots o_k$.

$z_t = \langle r, g, b, d \rangle$. Our observation consists of the pixel values and depth generated from a sensor on the robot's end effector. This sensor could be a depth camera from the Kinect; in our experiments we use a one-pixel depth camera created from Baxter's IR sensor and wrist camera.

$u_t = \{\langle x, y, z, r, p, y\rangle$ Alternately, the robot could attempt to grasp the object, leading to a new configuration of objects in the workspace.
m The map consists of a colored voxel map for each object.

We aim to jointly estimate a distribution over object poses along with the map, m:

$$p(x_t, m | z_0 \dots z_t, u_0 \dots u_t) \tag{2}$$

We formulate this problem as a Bayes' filter with a time update:

$$p(x_t, m | z_0 \dots z_{t-1}, u_0 \dots u_t) = \int p(x_t | x_{t-1}, u_t) \times p(x_{t-1}, m | z_0 \dots z_{t-1}, u_0 \dots u_{t-1}) dx_{t-1} \tag{3}$$

and a measurement update:

$$p(x_t, m | z_0 \dots z_t, u_0 \dots u_t) = \frac{p(z_t | x_t, m) P(x_t, m | z_0 \dots z_{t-1}, u_0 \dots u_t)}{p(z_t | z_0 \dots z_{t-1}, u_0 \dots u_t)} \tag{4}$$

A. Transition Model

Our transition model incorporates the effect of the control input on the robot's end effector as well as on all the objects:

$$p(x_t | x_{t-1}, u_t) = p(r_t, o_t^1 \dots o_t^k | r_{t-1}, o_{t-1}^1 \dots o_{t-1}^k, u_t) \tag{5}$$

We assume that the robot's end effector pose and the object poses are independent:

$$= p(r_t | u_t) \times p(o_t^1 \dots o_t^k | o_{t-1}^1 \dots o_{t-1}^k, u_t) \tag{6}$$

We also assume that each object is independent of the others; although in general we would like the robot to predict the effects of its grasps, in practice other objects can be moved by mistake. Therefore our transition model will have high uncertainty over all objects when the robot moves any object.

$$\approx p(r_t | u_t) \times \prod_k p(o_t^k | o_{t-1}^k, r_t) \tag{7}$$

We endow the robot with two types of control actions. In the first type, the robot plans collision free motions with conservative models of the boundaries of possible objects; these types of transitions move the robot's end effector but leave all the object poses unchanged. The second type is when the robot grasps an object. After an attempted grasp we assume the pose of all the objects have moved and need to be reestimated (in case the robot knocked an object during its attempted grasp)

B. Measurement Model

Our sensor consists of a depth camera; for simplicity we update one pixel at a time, although a sensor such as the Kinect would consist of many updates at once.

$$p(z_t | x_t, m) = p(z_t | r_t, o_t^1 \dots o_t^k, m) \tag{8}$$

When the object pose and map is known, we can identify the one object penetrated by the sensor:

$$p(z_t | x_t, m) = p(z_t | r_t, o_t^i, m^i) \tag{9}$$

We assume a sensor model with Gaussian noise on pixel values and depth values.

C. Inference

We carry out inference in the model using a Rao-Blackwellized Particle Filter [9]. In particular we partition our state space according to the product rule:

$$p(x_0 \dots x_t, m | z_0 \dots z_t, u_0 \dots u_t) = p(m | x_0 \dots x_t, z_0 \dots z_t) \times p(x_0 \dots x_t | z_0 \dots z_t, u_0 \dots u_t) \tag{10}$$

Thus each particle is represented by the set:

$$\left\{ w_t^{(i)}, x_0^{(i)} \dots x_t^{(i)}, p(m | x_0^{(i)} \dots x_t^{(i)}, z_0 \dots z_t) \right\} \tag{11}$$

In a simplified model, we can compute a maximum likelihood map analytically from the pose and observation history, rather than representing a distribution over it.

Our proposal distribution is the motion model, as in Fast Slam 1.0 [26]:

$$x_t^{(i)} \sim p(x_t | x_{t-1}^{(i)}, u_t) \tag{12}$$

That is, when objects do not move, we assume a fixed pose, together with small random movement to correct initial pose estimation error.

IV. Experimental Setup

JGO: This is where we describe the experiments we performed. We are not trying to extend the state-of-the-art on our individual tasks. Rather, we are providing an interactive framework which will raise the maximum automated vision available to the average user.

Our system can be evaluated in two important ways. Firstly, how effective is the system at the tasks for which it is trained? Secondly, how accessible to users is the system? For now we evaluate system performance, and leave user studies for future work.

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

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

B. Grasp Experiments

For grasp experiments, we conducted online trials in order to compute success rates. This uses grasp success rate as an objective function.

Expert annotation of grasps

Uniform grasp sampling

Thompson sampling

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.

C. PID Control Experiments

For PID experiments, we conducted online trials in order to compute success rates. We use the time to convergence as the objective function

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

Summary:

People doing SLAM. Wang et al. [36], Gallagher et al. [11],

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. [24], Goldfeder et al. [12]

Object tracking in vision (typically surveillance).

POMDPs for grasping. Platt et al. [28], Hsiao et al. [15]

People doing systems. Hudson et al. [16], Ciocarlie et al. [8]

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. [37] 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.

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.

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. [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.

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

Velez et al. [35] 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, 29] 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 [5, 25]. 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 [31, 12, 27]. 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 [6] **ST: Need a whole paragraph about that.**

ST: Ciocarlie et al. [8] 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 [23]. 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 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 [36, 11, 30, 32].

Our approach is similar to the philosphy adopted by Rethink Robotic's Baxter robot, and indeed, we use Baxter as our test platform [10]. **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 [4, 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 [33] 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.

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?

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] 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.

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

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

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

[8] 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.

[9] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part i. *Robotics & Automation Magazine, IEEE*, 13(2):99–110, 2006.

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

[11] 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.

[12] 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.

[13] 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.

[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] 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.

[24] 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.

[25] 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.

[26] Michael Montemerlo, Sebastian Thrun, Daphne Koller, Ben Wegbreit, et al. Fastslam: A factored solution to the simultaneous localization and mapping problem. In *AAAI/IAAI*, pages 593–598, 2002.

[27] 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.

[28] 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.

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

[30] 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.

[31] 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.

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

[33] 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.

[34] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT Press, 2008.

[35] 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.

[36] 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.

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