A Robot in the First Person

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 issue, we propose an architecture for perception and actuation that enables a robot to quickly and easily acquire instance-level descriptions of novel objects. The robot learns to recognize, estimate the pose, and grasp the object through active data collection as well as by asking a human annotator for specific supervision. Our approach converts the task of *category recognition* (pick up any mug) to *instance recognition* (pick up this mug), a much easier problem that conventional methods can solve with the right training data. By leveraging this first-hand experience with the object, naive users can quickly train our system to reliably localize and grasp the specific objects that they care about for their particular application. As a byproduct, our approach collects data on grasping and manipulation that we can use to train general-purpose models. 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. 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 [].

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 []. 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 [].

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. It is powerful because it learns to identify and grasp on a per object basis. It 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.

There are three parts to our approach. The first part is recognizing the category of an object and the the second part is estimating the pose of the object, both of which we accomplish with simple and robust computer vision algorithms. The third part 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.

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. 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 pick and place the target object.

Additionally, 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.

Do we also want to train the parameters of the PID controller? That is non-trivially cool as well.

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).

II. Related Works

Talk about each task, both at the level of sophistication we are using and at the state of the art. Address typical expert training and automated approaches that exist, together with usability concerns.

A. Object Recognition

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

B. Pose Estimation

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.)

C. Object Grasping

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

III. General Autonomous Training Algorithm

we teach the system by finding its weaknesses and training it to overcome these weaknesses. We provide a framework for structuring that training in an affordable, scalable way. we structure the training, the training structures the models.

"I see you're having trouble picking up gyroBowls. Lets think about it differently (initialize generic pose model) and practice gazing a bit (train the pose model)."

Also, grounding ambiguous object examples to the right class, retraining the models online.

IV. Applications of GATA

A. Recognition Training

B. Pose Estimation Training

C. Grasp Training

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.

4. Once aiming is complete, save the image in the reticle, try to grasp, and save whether that grasp completed. This can be ascertained by the gripper position.

5. Calculate a density map for this

D. PID Parameter Training

Maybe a coordinate descent algorithm or wide-scale random noise search.

V. Experimental Setup

A. Object Categorization

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. Pose Estimation

This should be analogous to categorization where the classes are now the poses.

C. Grasp Experiments

This uses grasp success rate as an objective function.

Expert annotation of grasps

Uniform grasp sampling

Thompson sampling

D. PID Experiments

We probably use the time to convergence as the objective function

VI. Evaluation

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?

Because this system is designed as a component in a complex system, raw recognition rates and pose estimation accuracy would not be very informative even if it were not impractical to obtain them. Besides, we are not trying to extend the state-of-the-art on those tasks. Rather, we are trying to provide an interactive framework which will raise the maximum automated vision available to the average user.

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. Expert Manual Training

We therefore 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. Laboratory Automatic Training

How well does the automatic training system perform when trained in laboratory conditions?

C. Non-Expert In-The-Wild Training

We then go on to compare the performance of the system when trained to various degrees by naive and technical non-expert users.

We repeat the experiments with naive users in two settings. In the first, they train laboratory objects. In the second, they provide their own objects.

[**?**]

VII. Extensions

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