Acquiring Object Experiences at Scale

John Oberlin and Maria Meier and Tim Kraska and Stefanie Tellex
Computer Science Department
Brown University

***Abstract*—The aim of this project is to improve the performance of automatic object detection, tracking and manipulation by using robots to collect a corpus of perceptual data for one million real-world objects. Robots lack the ability to robustly identify, localize, and manipulate the objects in our daily lives. The field of object detection and recognition is driven by annotated corpora [21, 8, 15, 26] which researchers use to train and test models [13, 10]. These corpora consist of photos taken by a human photographer and may contain many examples of objects, but typically only a single view of each individual object. Existing corpora of object instances contain many fewer examples (on the order of hundreds) and no experience interacting with the object [14, 23]. Our proposed approach, in contrast, uses an industrial robot (Baxter) to automatically collect a database of object models, including RGB images from multiple views, point clouds, and physical experiences gathered as the robot interacts with each object. Using our proposed approach, anyone with access to a Baxter robot can scan an object, acquiring a model of that object for their immediate use and also adding that model to the database. If we had all 300 research Baxters scanning and interacting with objects using our existing software stack, we could reach our goal of one million objects in eleven days. We are soliciting participation for our "Million Object Challenge" to revolutionize robotic manipulation by providing access to object experiences at a very large scale.**

I. Introduction

Robust object manipulation is essential for human-robot collaboration, because a robot needs to be able to manipulate a diverse array of objects to assist with a wide variety of tasks, such as helping a chef in a kitchen by delivering a spatula, a patient in a hospital by pouring them a glass of water, or a worker in a factory by handing them a screwdriver. Variations in the amount of clutter, lighting conditions, and properties of domain specific objects make it difficult to create a general-purpose system for manipulating diverse objects in real-world environments. State of the art computer vision approaches such as deep learning may begin to address these challenges, but require massive amounts of data which are usually not provided in a way that promotes joint learning of grasps [21, 8, 15]. Existing datasets of many objects usually contain only a few views of each object, and detailed datasets generally only contain examples for a few objects [14, 23]. Furthermore they lack information that will lead to the discovery of object affordances, such as unscrewing a bottlecap, or opening a book.

A two-year old human child is an effective mobile-manipulator because of hours of practice spent playing with countless objects [9]. To enable a robot to benefit from a similarly-sized dataset, we aim to collect a very large corpus of one million real-world object memories, containing information gathered both passively, such as RGB images and point clouds, as well as actively, such as memories about which grasps succeeded well enough to transport an object. To collect this corpus, we will use the Baxter robot, which consists of two seven-degree of freedom arms equipped with a monocular camera and IR range sensor in each hand. By exploiting Baxter's ability to move the camera and manipulate objects, the robot can collect data both passively, by imaging the object, and actively, by interacting with it in the environment. Baxter provides a unique advantage and opportunity for this project because there are around 300 research Baxters in the robotics labs of today. Our scanning framework can acquire an object in about 20 minutes using an unmodified Baxter robot that does not need to be augmented with any additional sensing. Using this framework, we could scan one million objects using this method in eleven days, if we had access to all 300 existing research Baxters. We propose to create the "Million Object Challenge" to enable people to collaborate to reach our goal of a million object instance models.

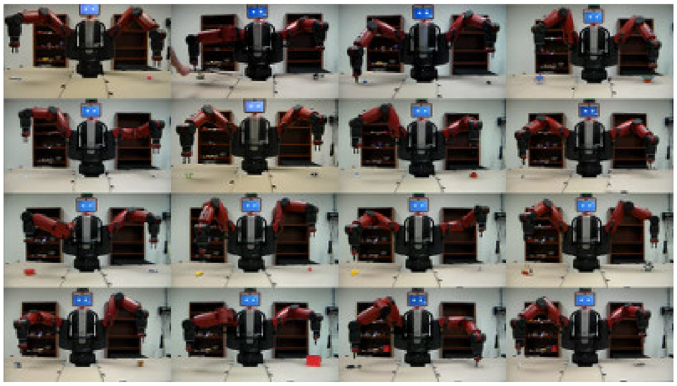


Fig. 1. We aim to automate the collection of large datasets of object manipulation results by distributing the load across many Baxter research robots.

This paper outlines our approach to acquiring object experiences on a very large scale. We describe our software framework for collecting data about objects both passively and actively using Baxter, described in more detail in our previous paper [18], adding an analysis of timing information and describing issues faced when scanning novel objects. Scenes from the data collection process appear in Figure 1, and a video is available at [1]. Next we discuss our plans to scale up data collection by creating a distributed database architecture for scanning objects and sharing data and present approaches for using this data to revolutionize robotic object recognition and manipulation by leveraging the information contained in this very large dataset.

II. Acquiring Object Appearance Data

We take an instance-based approach to object manipulation, training a model specific to that object so that we can detect, localize and grasp the object, in contrast to category-based approaches [22, 19]. Instance-based approaches do not generalize to novel objects, but work reliably with sufficient data. Our approach works by using the robot to automatically collect this data, scanning the object to build an instance-based detector specific to that object and proposing grasps using geometric information from the point cloud.

A. Object Detection

To detect a target object, the robot collects a dataset of image crops by moving its end effector around the target, takes images of the object, and then extracts a bounding box of the target from each image. A modified Canny edge detector [7] provides bounding boxes for objects in view, which allows us to perform coarse grained visual servoing on previously unencountered objects. To automatically extract clean bounding boxes, we use a structured environment consisting of a tabletop around the robot. The surface of the table is allowed to have some visual texture but should be perpendicular to the robot's z-axis. Reducing the dimensionality of the pose estimation subproblem, the robot keeps the camera pointed straight down at the table in what we call a crane pose, but the orientation within that plane changes during visual servoing in order to move into the correct frame.

To detect objects using this data, the robot uses a standard SIFT [16], bag of words (BoW) [24], and k-nearest neighbors (kNN) [4] pipeline. We extract SIFT features densely from image crops, cluster features from all object classes once at train time to form a visual vocabulary for the BoW model, then create a histogram BoW feature for each example at train time to build the kNN model and also at inference time to query against the kNN model. We employ machine learning techniques of relatively low power, which were considered state-of-the-art nearly a decade ago; they are effective because the robot can move the camera about the object to obtain a very clean image crop.

B. Pose Estimation

To localize the object, the robot obtains registered crops of a constant size at known heights, which it uses to perform fine grained visual servoing on the object. Fine grained servoing allows us to move the arm back the origin of the local coordinate frame of the object at the original registered orientation.

We perform object registration through visual servoing, for which we require constant size $W \times H$ crops $\{A_i\}$ of the object centered beneath the camera at known heights. At servo time, we extract a much larger crop C from the camera image and perform the modified Canny edge detection on it. Next we examine all of the $W \times H$ patches of C and for each patch $P_{x,y}$, subtract its mean and L^2 normalize it to yield $\hat{P}_{x,y}$, having performed the same procedure once to the registration crops $\{A_i\}$ to obtain $\{\hat{A}_i\}$. Selecting the normalized registration crop $\hat{A}_h$ from the appropriate height h, we apply $\hat{A}_h$ and a fixed subset $\{\hat{A}_{h,\theta}\}$ of its rotations to the normalized crops $\hat{P}_{x,y}$ and select the (x, y, θ) with maximal inner product $\langle \hat{A}_{h,\theta}, \hat{P}_{x,y} \rangle$. This triple describes the movement necessary to take us to the local frame of the object. This procedure localizes many objects to high-enough precision to grasp, but can fail due to specular effects on reflective objects.

C. Grasping

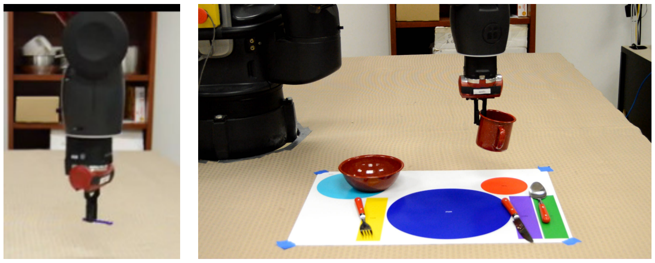
To propose grasp points, the robot collects a point cloud and finds regions that fit into its gripper. Since Baxter only contains a triangulation range finder located on the end effector, we perform a raster scan by moving the sensor over the object. This procedure is slow but enables us to collect 3D information using only the sensors that come with Baxter, lowering the barrier to entry to our scanning project. This map allows us to propose grasps on specific physical locations of the object measured in centimeters, to find the "center" of the object, and to determine its principle axes. The depth map is slow to build and usable maps can take over five minutes to construct, so we only build the map once and reuse it in the future by performing registration with RGB data alone.

A primary goal of developing this system is to gather data with minimum interaction from the user. Human time is valuable; we can make data collection less expensive by enabling the robot to gather data on its own. There is a tradeoff between data quality and autonomy. We can choose to annotate general 3D grasps by hand, which are likely of better quality than inferred crane grasps, but to do so requires additional operator time. Instead, we can choose to let the robot infer its own grasps, which takes much more robot time but considerably less attention from the operator.

D. Scanning Time

A basic scan collects 144 variable-size pose-annotated RGB image crops from a single height for detection, four constant size images from different heights for visual servo registration, and a raster IR depth map which can be used to infer grasp proposals. It takes less than one minute of human time and about 23 minutes of robot time to collect the data for the basic scan. The reinforcement learning step takes about 50 seconds per attempted grasp plus human time for any object resets.

A minimal scan collects the 144 RGB crops and the four crops for registration. This process takes less than one minute of human time and about five and a half minutes of robot time, but it requires the operator to annotate a grasp. Annotation requires less than 30 seconds to register the image plus about 30 seconds per grasp, and additional time to verify their quality. This annotation is necessary for us because Baxter's stock gripper is a parallel gripper and requires precise placement in order to grip an object. Using other grippers [17] may allow us to forgo annotation by always aiming at the center of the object; additionally we aim to explore approaches for category-based grasp proposals [22, 19].



(a) Picking a snap circuit part.

(b) Setting the table using YCB objects.

Fig. 2. Our robot picking various objects.

Our system enables a quick turn-around between seeing a new object and picking. For example, after seeing a talk on picking Snap Circuit pieces [11], we used our system to acquire a model and pick one of the parts, shown in Figure 2(a). We recently received the new YCB dataset [6], which contains nearly one hundred distinct objects for use in benchmarking robotic grasping. To benchmark our system, we performed the table setting task, which involves grasping and placing a mug, bowl, plate, fork, spoon, and knife in a predetermined place setting. Our score on the benchmark was ten: the robot was able to move five of the objects to the appropriate place on the table although it did not place them precisely inside the goal region. To achieve this score, we scanned the six objects in less than 60 minutes of robot time and a few minutes of person time. After scanning, we could detect and manipulate the objects individually, with the exception of the plate, which does not fit into Baxter's fairly restrictive kinematic workspace. When the fork, spoon, and knife were in the workspace at the same time, the robot often confused them, because these objects are highly reflective on their distinct regions. Despite the confusion, manipulation was still fairly successful, because all utensils had similar geometry and grasp points. Figure 2(b) shows the robot picking the YCB objects.

III. Acquiring Object Manipulation Experiences

Passive data collection by imaging the object provides a powerful capability to detect and manipulate. However, much richer data can be obtained by actively experimenting with the object and learning about how it changes in response to the robot's actions. Moving the object can enable the robot to learn how to pick it up, how its appearance changes in different poses, what its stable poses are, and how to most reliably detect, localize, and grasp the object. By focusing on an instance-based approach, the robot can achieve high reliability for specific objects, but also record its experiences to create a corpus for later generalization of its knowledge to new objects.

A. Acquiring Grasp Experiences

To autonomously learn how to pick an object, we take the bandit-based approach in our previous work and summarized here [18]. We consider crane grasps described by (x, y, θ) triples, where (x, y) lies on a grid of one centimeter spacing surrounding the object and θ is a grasp angle of either 0, $\frac{\pi}{4}$, $\frac{\pi}{2}$, or $\frac{3\pi}{4}$. Correlation with hand designed 3×3 filters yields grasp proposals which are often successful but frequently not. To clean up the proposals, we frame the problem of learning grasps as a multi-armed bandit problem, where each (x, y, θ) triple corresponds to an arm. We saw a time improvement over Thompson sampling [25, 3] in our case with our own confidence bound based algorithm. When using the top proposal grasp for each object, our system succeeded $5/10$ times on average, but succeeded $0/10$ times for some of our objects. After reinforcement learning with our confidence bound algorithm, pick success was $7.5/10$ on average and non-zero for all of our reported objects [18].

We performed our full basic scan and reinforcement learning for 30 objects. Before reinforcement learning, the grasp proposal success rate was $5/10$ on average. After learning, the rate increased to $7.5/10$ and many objects went from zero to non-zero success.

B. Acquiring Pose Experiences

A key problem with our active grasping process manifests when the robot knocks an object over during an interaction. Because we are using instance-based models, when the robot drops an object to a new stable pose, it can no longer detect, localize or predict grasps. To address this problem we aim to automatically detect the gravitationally stable poses of an object and discover how to transition between them through manipulation with its gripper and more generally with its environment. We aim to acquire a model of the object that allows the robot to learn how to move an object between stable modes from experience, by actively grasping the object and rotating it to new positions. This framework will enable the robot to learn that a cup can be place upright, on its side, or upside-down, and learn to plan manipulation actions to return a cup to its canonical orientation.

IV. Sharing and Generalizing Object Experiences

Our goal is to enable researchers around the world to contribute to our scanning project by installing software on their robot, acquiring object models for pick-and-place at their site, but also contributing to our effort to collect a corpus of one million objects.

A. Sharing Object Experiences

To share object experiences at the scale of millions of objects, it is necessary to have a database to collect the information. In our current framework, the robot stores all data it has collected about the object in a folder as images and YAML [2] files, including the point cloud, registered overhead views, and RGB crops. After each scan, our software uploads the object data to a central database. This database server supports automatic upload, download, and searching of the data. In this way researchers around the world can contribute to our project by installing our software, scanning objects, and retrieving models scanned by others. Figure 3 shows a

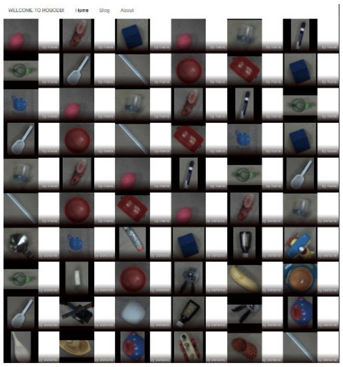


Fig. 3. Screenshot from our website that serves scanned objects.

screenshot from our website where scanned models can be downloaded.

Transferring models across robots is challenging. When the same system produces and uses a model, small errors in algorithms and calibration can go unnoticed; a single system in isolation need not be accurate if it is precise. Our system can automatically calibrate itself with minimal interaction from the user, who need only supply a structured working surface, which is allowed a generous amount of visual texture. Using only items that come with Baxter and common household goods, the robot can be calibrated and scanning objects in less than an hour after ROS [20] is installed. Our software stack is a single executable with few dependencies outside of ROS and OpenCV [5], making it easy to install and deploy. We have deployed our software on one other Baxter, where it was able to scan objects and pick and we plan to scale up in the coming months.

B. Generalizing Object Experiences

Our longer-term aim is to enable the robot to generalize its experiences by exploiting this very large dataset. We aim to predict general grasps from visual experience by using our corpus as a training set of successful grasps paired with images as the gripper approaches the object. Similarly we can improve the robot's ability to segment objects into parts (such as handles and lids) and predict unseen components from a single view by generalizing from the multi-view data. This dataset will open up a variety of new research tasks by leveraging state-of-the-art computer vision techniques [13] to predict the observed effects of the robots actions, such as grasp success, pose estimation accuracy, or specular effects.

V. Conclusion

We propose the "Million Object Challenge," a collaborative effort to create a new dataset of objects larger than any existing dataset of its type by four orders of magnitude; this dataset will transform robotic manipulation, as well as object detection and localization in images and video, whether it is surveillance footage or videos on YouTube. Our system is functional and easy to use on a small scale. It enables us to perform studies on human to robot interaction and is a proof of concept which demonstrates that this technology could be deployed for stocking and retrieval tasks.

We will enable the robot to actively explore the object to learn its affordances, such as taking the top off of a bottle to pour water out of it, the ability of a box to hold objects, or the fact that the lights turn on when a switch is flicked. Attacking this problem currently requires careful definitions and hand structured models. We hope to make this process automatic by teaching states and affordances to the robot and enabling it to construct POMDPs [12] over those values through observation and interaction with the world.

We aim to extend our approach to a mobile-manipulator robot, enabling the robot to actively explore complex environments and learn affordances for objects such as doors, light switches, and cupboards. A mobile robot can also learn about objects by placing them in different locations and remembering the ground truth pose, so it can actively detect appearance information in different lighting conditions and clutter.

More generally our instance-based mapping approach to objects and environments allows a robot to actively explore through trial and error and self-annotate its world by remembering the effects of its own actions. Leveraging this self-annotation allows the robot to remember and generalize from its own experiences.

Acknowledgments

We would like to thank Bianca Homberg and Daniela Rus for their help installing our software stack on their Baxter.

References

[1] Bandit-Based Adaptation for Robotic Grasping. https://www.youtube.com/watch?v=xfH0B3g782Y. Accessed: 2015-05-31.

[2] The Official YAML Web Site. http://yaml.org/. Accessed: 2015-05-31.

[3] Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. *JMLR: Workshop and Conference Proceedings*, 23, 2012.

[4] N. S. Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.

[5] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[6] Berk Calli, Aaron Walsman, Arjun Singh, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. Benchmarking in manipulation research: The ycb object and model set and benchmarking protocols. *arXiv preprint arXiv:1502.03143*, 2015.

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

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[9] Lise Eliot. Whats going on in there. *How the brain and mind develop in the first five years of life*, pages 237–239, 1999.

[10] Mark Everingham, Luc Gool, Christopher K. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88:303–338, June 2010. ISSN 0920-5691.

[11] Bradley Hayes and Brian Scassellati. Developing effective robot teammates for human-robot collaboration. In *2014 AAAI Fall Symposium Series*, 2014.

[12] Leslie Pack Kaelbling and Tomas Lozano-Perez. Hierarchical planning in the now. In *Workshops at the Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[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] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014*, pages 740–755. Springer, 2014.

[16] 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.

[17] Raymond R Ma, Lael U Odhner, and Aaron M Dollar. A modular, open-source 3d printed underactuated hand. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 2737–2743. IEEE, 2013.

[18] John Oberlin and Stefanie Tellex. Autonomously acquiring instance-based object models from experience. 2015. Under review.

[19] Andreas ten Pas and Robert Platt. Localizing grasp affordances in 3-d points clouds using taubin quadric fitting. *arXiv preprint arXiv:1311.3192*, 2013.

[20] Morgan Quigley, Ken Conley, Brian P. Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software*, 2009.

[21] Bryan C Russell, Antonio Torralba, Kevin P Murphy, and William T Freeman. Labelme: a database and web-based tool for image annotation. *International journal of computer vision*, 77(1-3):157–173, 2008.

[22] 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.

[23] Arjun Singh, James Sha, Karthik S Narayan, Tudor Achim, and Pieter Abbeel. Bigbird: A large-scale 3d database of object instances. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 509–516. IEEE, 2014.

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

[25] 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.

[26] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1521–1528. IEEE, 2011.