A Dataset for Grasping and Manipulation using ROS

Matei Ciocarlie[†], Gary Bradski[†], Kaijen Hsiao[†] and Peter Brook[†*]

***Abstract*— In this study, we introduce a household object dataset for recognition and manipulation tasks, focusing on commonly available objects in order to facilitate sharing of applications and algorithms. The core information available for each object consists of a 3D surface model annotated with a large set of possible grasp points, pre-computed using a grasp simulator. The dataset is an integral part of a complete Robot Operating System (ROS) architecture for performing pick and place tasks. We present our current applications using this data, and discuss possible extensions and future directions for shared datasets for robot operation in unstructured settings.**

I. Datasets for Robotics Research

Recent years have seen a growing consensus that one of the keys to robotic applications in unstructured environments lies in collaboration and reusable functionality. An immediate result has been the emergence of a number of platforms and frameworks for sharing operational "building blocks," usually in the form of code modules, with functionality ranging from low-level hardware drivers to complex algorithms such as path or motion planners. By using a set of now well-established guidelines, such as stable documented interfaces and standardized communication protocols, this type of collaboration has accelerated development towards complex applications. However, a similar set of methods for sharing and reusing data has been slower to emerge.

In this paper we describe our effort in producing and releasing to the community a complete architecture for performing pick-and-place tasks in unstructured (or semi-structured) environments. There are two key components to this architecture: the algorithms themselves, developed using the Robot Operating System (ROS) framework, and the knowledge base that they operate on. In our case, the algorithms provide abilities such as object segmentation and recognition, motion planning with collision avoidance, grasp execution using tactile feedback, *etc*. The knowledge base, which is the main focus of this study, contains relevant information for object recognition and grasping for a large set of common household objects.

Some of the key aspects of combining computational tools with the data that they operate on are:

- other researchers will have the option of directly using our dataset over the Internet (in an open, read-only fashion), or downloading and customizing it for their own applications;
- defining a stable interface to the dataset component of the release will allow other researchers to provide their own modified and/or extended versions of the data to the community, knowing that it will be directly usable by anyone running the algorithmic component;
- the data and algorithm components can evolve together, like any other components of a large software distribution, with well-defined and documented interfaces, version numbering and control, *etc*.

In particular, our current dataset is available in the form of a relational database, using the SQL standard. This choice provides additional benefits, including optimized relational queries, both for using the data on-line and managing it off-line, and low-level serialization functionality for most major languages. We believe that these features can help foster collaboration as well as provide useful tools for benchmarking as we advance towards increasingly complex behavior in unstructured environments.

There have been previous example of datasets released in the research community (as described for example in [3], [7], [13] to name only a few), used either for benchmarking or for data-driven algorithms. However, few of these have been accompanied by the relevant algorithms, or have offered a well-defined interface to be used for extensions. The database component of our architecture was directly inspired by the Columbia Grasp Database (CGDB) [5], [6], released together with processing software integrated with the *GraspIt!* simulator [9]. The CGDB contains object shape and grasp information for a very large ($n = 7,256$) set of general shapes from the Princeton Shape Benchmark [12]. The dataset presented here is smaller in scope ($n = 180$), referring only to actual graspable objects from the real world, and is integrated with a complete manipulation pipeline on the PR2 robot.

II. The Object and Grasp Database

A. Models

One of the guiding principles for building this database was to enable other researchers to replicate our physical experiments, and build on our results. The database was constructed using physical objects that are generally available from major retailers (while this current release is biased towards U.S.-based retailers, we hope that a future release can include international ones as well). The objects were divided into three categories: for the first two categories, all objects were obtained from a single retailer (IKEA and Target, respectively), while the third category contained a set of household objects commonly available in most retail stores. Most objects were chosen to be naturally graspable using a single hand (*e.g.* glasses, bowls, cans, *etc.*); a few were chosen as use cases for two-hand manipulation problems (*e.g.* power drills).

[†]Willow Garage Inc., Menlo Park, CA. Email: {matei, bradski, hsiao, pbrook}@willowgarage.com
[*]University of Washington, Seattle, WA.

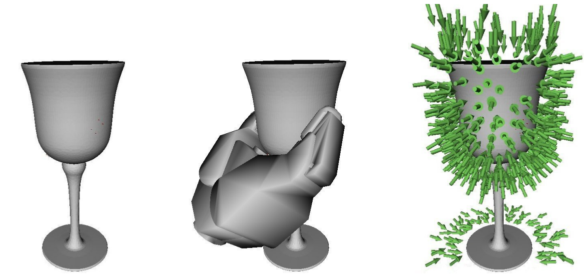

Fig. 1. Grasp planning in simulation on a database model. Left: the object model; Middle: grasp example using the PR2 gripper; Right: the complete set of pre-computed grasps for the PR2 gripper.

For each object, we acquired a 3D model of its surface (as a triangular mesh). To the best of our knowledge, no off-the-shelf tool exists that can be used to acquire such models for a large set of objects in a cost- and time-effective way. To perform the task, we used two different methods, each with its own advantages and limitations:

- for those objects that can be described as surfaces of rotation, we segmented a silhouette of the object against a known background, and used rotational symmetry to generate a complete mesh. This method can generate high-resolution, very precise models, but is only applicable to rotationally symmetrical objects.
- for all other objects, we used the commercially available tool 3DSOM (Creative Dimension Software Ltd., U.K.). 3DSOM builds a model from multiple object silhouettes, and can not resolve object concavities and indentations.

We believe that the 3D models provided in the current release can be useful for a number of algorithms and tasks (as we will exemplify in this paper). For future releases, we are investigating alternative methods for building models of more general objects, such as combining multiple high-resolution stereo images.

Overall, for each object, the database contains the following core information:

- the maker and model name (where available);
- the product barcode (where available);
- a category tag (*e.g.* glass, bowl, *etc.*);
- a 3D model of the object surface, as a triangular mesh.

B. Grasps

For each object in the database, we used the *GraspIt!* simulator to compute a large number of grasp points for the PR2 gripper. We note that, in our current release, the definition of a good grasp is specific to this gripper, requiring both finger pads to be aligned with the surface of the object and further rewarding postures where the palm of the gripper is close to the object as well. Our grasp planning tool used a simulated annealing optimization, performed in simulation, to search for gripper poses relative to the object that satisfied this quality metric. For each object, this optimization was allowed to run over 4 hours, and all the grasps satisfying our requirements were saved in the database; an example of this process is shown in Fig. 1. This resulted in an average of 600 grasp points for each object. In the database, each grasp contains the following information:

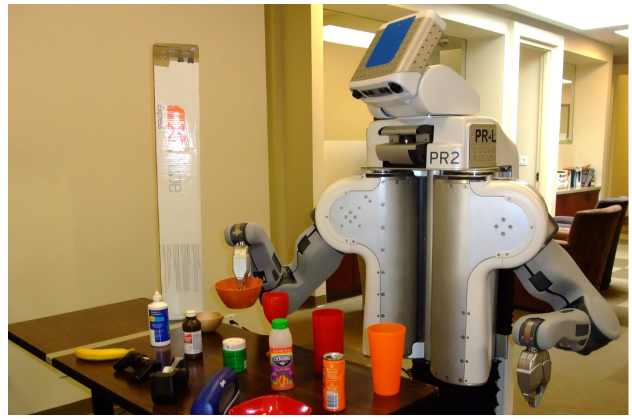

Fig. 2. The PR2 robot performing a grasping task on an object recognized from the model database.

- the pose of the gripper relative to the object;
- the value of the gripper degree of freedom, determining the gripper opening;
- the value of the quality metric used to distinguish good grasps.

We note that this process can easily be extended to other robot hands as well. For more dexterous models, a different grasp quality metric can be used, taking into account multi-fingered grasps, such as metrics based on the Grasp Wrench Space [4]. The Columbia Grasp Database also shows how large scale off-line grasp planning is feasible even for highly dexterous hands, with many degrees of freedom [5]. We hope that future releases will also include grasp information for some of the robotic hands most commonly used in the research community.

III. Applications

The database described in this study was integrated in a complete architecture for performing pick and place tasks on the PR2 robot. A full description of all the components used for this task is beyond the scope of this study. We present here a high-level overview with a focus on the interaction with the database; for more details on the other components we refer the reader to [2].

In general, a pick-and-place task begins with a sensor image of the object(s) to be grasped, in the form of a point cloud acquired using a pair of stereo cameras. Once an object is segmented, a recognition module attempts to find a match in the database, using an iterative matching technique similar to the ICP algorithm. We note that this recognition method only uses the 3D surface models of the objects stored in the database. In the future, we intend to experiment with more powerful object recognition methods, which might require different types of data for each database object.

If a match is found between the target object and a database model, a grasp planning component will query the database for all pre-computed grasp points of the recognized object. Since these grasp points were pre-computed in the absence of other obstacles and with no arm kinematic constraints, an additional module checks each grasp for feasibility in the current environment. Once a grasp is deemed feasible, the motion planner generated an arm trajectory for achieving the grasp position, and the grasp is executed. An example of a grasp executed using the PR2 robot is shown in Fig. 2.

Even though this study is mostly concerned with the known object dataset, we believe it is important to point out that this manipulation pipeline can also operate on novel objects. In this case, the database-backed grasp planner is replaced by an on-line planner able to compute grasp points based only on the perceived point cloud from an object. Grasp execution for unknown objects is performed using tactile feedback in order to compensate for unexpected contacts. We believe that a robot operating in an unstructured environment should be able to handle unknown scenarios while still exploiting high-level perception results and prior knowledge when these are available. This dual ability also opens up a number of promising avenues for autonomous exploration and model acquisition which we will discuss in the next section.

In an experiment designed to test the reliability of the grasping architecture, the robot was to perform 2 pick-and-place operations on each of 15 objects, for a total of 30 operations. 29 of 30 operations succeeded; one object was inadvertently collided with while executing a different task. 10 of the 15 objects were in the model database; in 15 of 20 detections they were correctly recognized. In addition, 2 unknown objects were mistakenly classified as database models. However, of these 7 recognition errors, only 2 resulted in grasp failures: 3 objects were grasped successfully even though they were not recognized (being treated as novel objects), and 2 were recognized as models that were close enough in shape to the true object to allow task completion. Finally, one object was dropped despite correct recognition. This behavior was also demonstrated in a live environment at the 2010 Intl. Conf. on Robotics and Automation. Over three days of execution, the system often operated continuously without grasp failures for periods of time ranging from 30 minutes to 1 hour, and successfully grasped novel objects (such as shoes, apples, keys, or hats) supplied by spectators.

IV. Work in Progress and Future Directions

We believe that the dataset that we have introduced, while useful for achieving a baseline for reliable pick and place tasks, can also serve as a foundation for more complex applications. Efforts are currently underway to:

- improve the quality of the dataset itself;
- improve the data collection process, aiming to make it faster, less operator-intensive, or both;
- use the large computational budgets afforded by off-line execution to extract more relevant features from the data, which can in turn be stored in the database;
- develop novel algorithms that can make use of this data at runtime;
- improve the accessibility and usability of the dataset for the community at large.

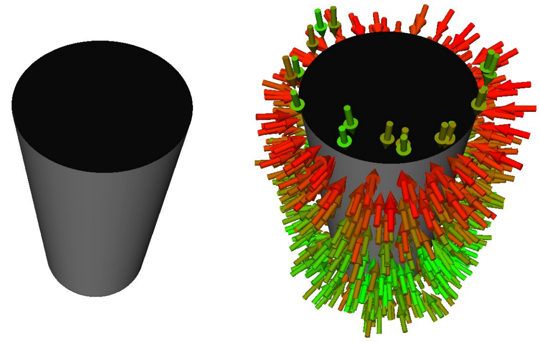

Fig. 3. (Best seen in color) Quantifying grasp robustness to execution errors, from low (red markers) to high (green markers). Note that grasps in the narrow region of the cup are seen as more robust to errors, as the object fits more easily within the gripper.

One option for automatic acquisition of high quality 3D models for a wide range of objects is to use high-resolution stereo data, able to resolve concavities and indentations, in combination with a pan-tilt unit. Object appearance data can be extended to also contain 2D images, from a wide range of viewpoints. This information can then be used to pre-compute relevant features, both two- and three-dimensional, such as SURF [1], PFH [10] or VFH [11]. This will enable the use of more powerful and general object recognition methods.

The grasp information contained in the database can be exploited to increase the reliability of object pickup tasks. An example of relevant off-line analysis is the study of how each grasp in the set is affected by potential execution errors, stemming from imperfect robot calibration or incorrect object recognition or pose detection. Our preliminary results show that we can indeed rank grasps by their robustness to execution errors; an example is shown in Fig. 3. In its current implementation, this analysis is computationally intensive, but it can be performed off-line and the results stored in the database for online use.

The grasping pipeline presented here can also serve as a foundation for fully automatic model acquisition: a robot can grasp a previously unseen object, inspect it from multiple viewpoints, and acquire a complete model, using techniques such as the ones presented in [8]. Additional meta-data, such as object classes or labels, can be obtained through on-line tools such as Mechanical Turk (Amazon.com, U.S.A.). This is a particularly compelling application, where multiple robots can independently augment the dataset, and share the results for increased versatility in manipulating a wider range of objects.

Finally, as the complete architecture described in this study is released as part of ROS, we hope that the research community will find new ways to use and extend this dataset. By making use of data integrated from multiple sources and environments, autonomous robots can potentially achieve higher reliability in more varied scenarios, as we strive towards truly robust performance in unstructured settings.

REFERENCES

[1] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. *Computer Vision and Image Understanding*, 110(3), 2008.

[2] M. Ciocarlie, K. Hsiao, E.G. Jones, S. Chitta, R. B. Rusu, and I. A. Sucan. Towards reliable grasping and manipulation in household environments. In *RSS Workshop on Strategies and Evaluation for Mobile Manipulation in Household Environments*, 2010.

[3] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In *CVPR Workshop on Generative-Model Based Vision*, 2004.

[4] C. Ferrari and J. Canny. Planning optimal grasps. In *IEEE International Conference on Robotics and Automation*, pages 2290–2295, 1992.

[5] C. Goldfeder, M. Ciocarlie, H. Dang, and P. Allen. The Columbia grasp database. In *ICRA*, 2009.

[6] C. Goldfeder, M. Ciocarlie, J. Peretzman, H. Dang, and P. Allen. Data-driven grasping with partial sensor data. In *IROS*, 2009.

[7] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007.

[8] M. Krainin, P. Henry, X. Ren, and Dieter Fox. Manipulator and object tracking for in hand model acquisition. In *ICRA*, 2010.

[9] Andrew Miller and Peter K. Allen. GraspIt!: a versatile simulator for robotic grasping. *IEEE Rob. and Autom. Mag.*, 11(4), 2004.

[10] R. B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (FPFH) for 3d registration. In *ICRA*, 2009.

[11] R. B. Rusu, G. Bradski, R. Thibaux, and John Hsu. Fast 3d recognition and pose using the viewpoint feature histogram. In *IROS*, 2010.

[12] Philip Shilane, Patrick Min, Michael Kazhdan, and Thomas Funkhouser. The Princeton shape benchmark. In *Shape Modeling and Applications*, 2004.

[13] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: a large dataset for non-parametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11), 2008.