Cloud-Based Robot Grasping with the Google Object Recognition Engine

Ben Kehoe[1] Akihiro Matsukawa[2] Sal Candido[4] James Kuffner[4] Ken Goldberg[3]

***Abstract*— Due to constraints on power and cost, robots operating in unstructured environments such as homes or offices have limits on onboard computation and limited data on the objects they encounter. *Cloud Robotics* proposes a thin-client model where robots are connected to modern cloud-computing infrastructure for access to distributed computing resources and datasets, and the ability to share training and labeling data for robot learning.**

We present a system architecture, implemented prototype, and initial experimental data on a cloud robotics system for recognizing and grasping common household objects. The prototype system incorporates a Willow Garage PR2 robot with onboard color and depth cameras, the Google Goggles image recognition system, the Point Cloud Library (PCL) for pose estimation, Columbia University's GraspIt! toolkit for grasp generation, and OpenRAVE for grasp selection. We extend our prior approach to sample-based grasp analysis to address 3D pose uncertainty and report timing and failure rates for experiments in recognition, pose estimation, and grasping.

I. Introduction

One as-yet unachieved goal of robotics and automation is an inexpensive robot that can reliably declutter floors, tables, and desks by identifying objects, grasping them, and moving them to appropriate destinations such as shelves, cabinets, closets, or trashcans. Errors in object recognition can be costly: an old chocolate bar could be mistaken for a cellphone and moved to the charging station, or vice versa—a cellphone could be placed in the trashcan. The set of objects that may be encountered in an unstructured environment such as a home or office is essentially unbounded and dynamically grows as our increasingly capitalist global economy designs new products to satisfy demand from consumers (and shareholders).

The cloud—the internet and its associated data and users—is a vast potential source for computation and data about objects, their semantics, and how to manipulate them. Users upload millions of digital photos every day and there are several image labeling projects using humans and machine learning [33] [36] [38]. We propose an architecture that integrates Google's image recognition system with a sampling-based grasping algorithm to recognize and grasp objects.

[1]Department of Mechanical Engineering; benk@berkeley.edu
[2]Department of Electrical Engineering and Computer Science; amatsukawa@berkeley.edu
[3]Department of Industrial Engineering and Operations Research and Department of Electrical Engineering and Computer Science; goldberg@berkeley.edu
[1–3] University of California, Berkeley; Berkeley, CA 94720, USA
[4]Google; 1600 Amphitheatre Parkway, Mountain View, CA 94043, USA; {scandido, kuffner}@google.com

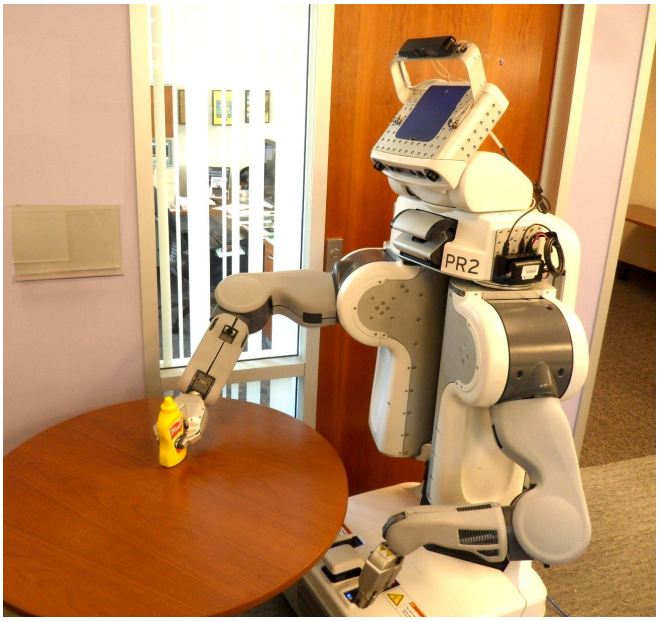

Fig. 1. In the prototype implementation, a cloud-connected PR2 sends onboard camera and depth sensor data to a Google server that performs object recognition and training and returns a set of grasps with associated confidence values.

Although networked robotics has a long history [3], Cloud Computing is a powerful new paradigm for massively parallel computation and real-time sharing of vast data resources. Cloud Robotics has the potential to significantly improve robots working in human environments in at least four ways: 1) indexing vast libraries of annotated image and object models with information on physical interactions [10], 2) massively-parallel sample-based motion planning and uncertainty modeling [7], 3) sharing of outcomes, trajectories, and dynamic control policies for commonly-used robot mechanisms such as Willow Garage's PR2 [39], 4) obtaining on-demand human guidance when needed [9] [35]. This paper exploits the first aspect for object recognition and the second aspect for grasp analysis.

In previous work, we have shown the utility of the cloud for computing grasps in the presence of shape uncertainty [21] [20]. In this paper, we extend that approach to consider objects with uncertainty in three dimensions. We train an object recognition server on a set of objects and tie it to a database of CAD models and candidate grasp sets for each object, where the candidate grasp sets are selected using the quality measure from our previous work. After the robot uses the object recognition service, the reference model is used to perform pose estimation of the object using the robot's 3D

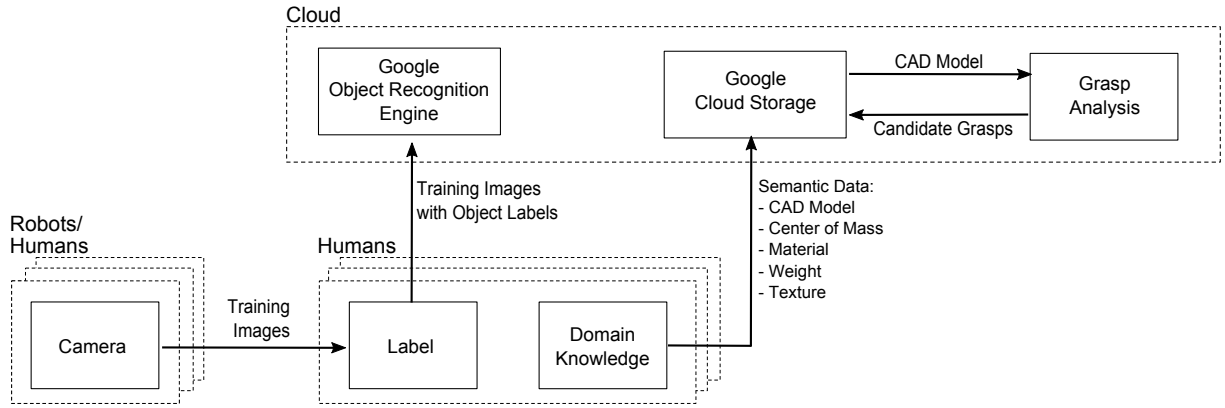

Fig. 2. System Architecture for offline phase. Digital photos of each object are recorded to train the object recognition server. A 3D CAD model of each object is created and used to generate a candidate grasp set. Each grasp is analyzed with perturbations to estimate robustness to spatial uncertainty.

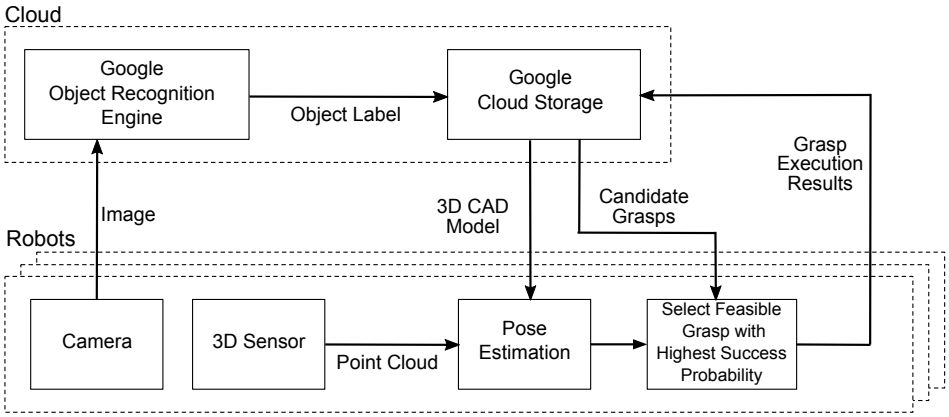

Fig. 3. System Architecture of online phase. A photo of the object is taken by the robot and sent via the network to the object recognition server. If successful, the server returns the stored data for the object. The robot then uses the detected point cloud with the 3D CAD model to perform pose estimation, and selects a grasp from the reference set of candidate grasps. After attempting the grasp, the results are stored in the cloud for future reference.

sensors, and one of the pre-computed grasps is selected for execution.

We evaluate our system on the Willow Garage PR2 [6]. The object recognition server was trained on 241 images (taken by the PR2) of a set of six household objects that represent common object shapes, with textual labels to aid the OCR facet of the object recognition algorithm. For each object, we created a reference model and a set of candidate grasps, which were uploaded to the server. Given this set of objects and associated grasp strategies, we demonstrate that (1) the robot is able to grasp the object based on pose estimation from stored reference data, and (2) none of the image processing or grasp selection is performed onboard the robot. The aggregation and sharing of training data means that training on 10 robots can theoretically occur 10 times as fast as training on a single robot.

II. Related Work

Object recognition is a very well-studied topic in computer vision, and there has been significant progress in many aspects of the problem, from the design of features that are invariant to translation, scaling, and rotation [24], to models for the problem [31], as well as links to other fields [12].

Researchers are working to improve both the scalability and accuracy of large-scale image recognition [19] [29] [30], making object recognition systems commercially viable. The purpose of this paper is to show how such a high-quality large-scale object recognition server can be incorporated into part of a cloud-based pipeline to improve grasping in robotics.

Recent research has demonstrated the cloud's ability to enable information sharing between networked robots to accomplish tasks widely separated in time and space [22] [25] [39], including work on object recognition using Google's 3D Warehouse [23]. In addition, computational tasks (such as object recognition) can be offloaded from robots into the cloud [7]. Finally, the cloud enables databases for robots to reuse previous computations in later tasks [10], which is especially useful for grasping [9] [15] [16] [28].

There is substantial research on grasping [8] While some research has looked at object recognition for grasping in isolation [18] [37], most work approaches it as a connected task. Approaches for object recognition for grasping include using local descriptors based on training images [11], and 3D model reconstruction involving 3D object primitives for pose estimation [17]. Saxena et al. [34] developed a method for calculating grasp points for objects based on images, where the grasps were learned from prior grasps for similar objects. This approach removed the need for a full 3D reconstruction of the object, but doesn't take advantage of

existing commercial object recognition systems.

Furthermore, the effectiveness of off-loading the computationally demanding task of large-scale image recognition to the cloud has also been explored for mobile augmented reality [14].

III. Problem Statement

As a prototype for a future system that incorporates many objects and labelled images, we consider a controlled experiment with a small set of objects and associated training and semantic data. A single object from the set is placed within the workspace of the robot. Using onboard camera and 3D scanner, the robot captures data about the object and connects to a server in the cloud to request identification and grasping strategy. Based on the responses, the robot attempts to grasp and move the object to an appropriate location.

Several preliminary setup steps occur in an offline phase. A set of sample objects that can be grasped by the robot end effector is selected on the basis of being reasonably distinct in shape and textural surface features including color and text. The size of the set, six objects in our experiments, is small because we have to manually capture images and models for each object.

Semantic information about each object is stored such as: object name, identifier, weight, surface properties such as friction, reference point sets, and 3D CAD model. For each object, we capture a representative set of digital images from different viewpoints and use these to train the Google Goggles object recognition system. We also pre-compute a set of robust grasp configurations for the PR2 robot and each object using the Columbia University GraspIt! toolkit. We use sampling of spatial perturbations of each object in stable resting configurations to estimate robustness of each grasp to spatial uncertainty.

Experiments are performed in the online phase, where the robot system detects an object and uses an onboard digital camera and 3D scanner to capture a 2D image of the object and a 3D point cloud. The robot sends the 2D image to servers in the cloud and retrieves an object identifier, confidence measure, semantic information about the object including a 3D CAD model and reference point set, or a report that the object is not recognized.

If the object is recognized, the robot system locally estimates the position and orientation (pose) of the object using the 3D CAD model, the reference 3D point set, and the measured 3D point cloud. The pose is then used to index the highest probability grasp from the cloud. The robot then transforms the grasp to the local environment and attempts to execute the grasp. Failure can occur due to false or incorrect identification, or after an unsuccessful grasp.

IV. System Architecture

As noted above, the system has two phases: offline and online. The system architecture of the offline phase is illustrated in Figure 2. This phase includes training of the object recognition server and the creation of object reference data as described in Section IV-A and the creation and analysis of the candidate grasp set as described in Section IV-B.

The system architecture of the online phase is illustrated in Figure 3. This phase starts when an object is detected by the robot system which takes a photo and captures a 3D point cloud and sends this to the object recognition server. The pose estimation and grasping is described in section Section IV-C.

A. Offline and Online Phases: Object Data

Fig. 4. A landmark photo taken by a mobile device is uploaded to the Goggles server, analyzed, and a recognition result is returned to the user.

Google Goggles is a popular network-based image recognition service accessible via a free downloadable app for mobile devices [2]. Users snap a photo of an unknown object or landmark and use the app to upload the photo which rapidly analyzes it to return a ranked list of descriptions and associated web links or a report that no reference can be identified (Figure 4).

In our prototype, we use a custom version of this system that runs on Google's production infrastructure, and is exposed as two HTTP REST [13] endpoints—one for training, and one for recognition. The training endpoint accepts 2D images of objects with labels identifying the object. The recognition endpoint accepts an image, and based on the set of features, either returns the object's identifier along with a probability of correctness, or reports failure.

The 3D CAD model, reference point set, and candidate grasp sets are hosted on Google Cloud Storage [1], which is a multi-tenant and widely-replicated key-value store. Each object's data is associated with the same unique string used to train the Goggles server. From this key, a REST URL can be constructed to retrieve the data.

The system first submits an image to the Google Goggles server to retrieve the object's identifying string, and then queries Cloud Storage for the reference data.

We associate a 3D CAD model with each object in the set. While the envisioned system would use the design models for the objects used, we generate models for our object set using 3D sensors. We first create a point cloud of the object using multiple Microsoft Kinects to scan all sides of the object at once. This point cloud is filtered using tools from PCL, the Point Cloud Library [5], and becomes our reference point set for the object. The reference point cloud is then used to create a 3D CAD model by using the surface reconstruction tools in PCL.

B. Offline Phase: Robust 3D Grasp Analysis

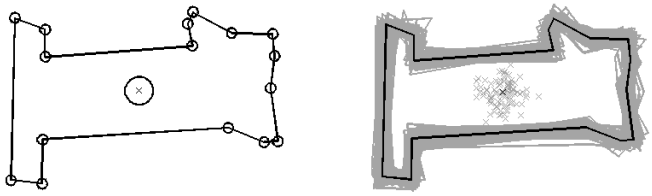

Fig. 5. Illustration of Shape Uncertainty Model and Sampled Shape Perturbations for 2D model. In this paper we generalize the sampling-based approach to 3D models with pose uncertainty.

The candidate grasp sets are generated using the Columbia University GraspIt! system [27]. We use a variant of our prior sampling-based algorithm for robustness to 2D shape uncertainty [21] and [20]. In that work, we assume a Gaussian uncertainty distribution for each vertex and center of mass (see Figure 5), and use 2D geometric features to determine if the gripper can push the object into a stable orientation. The algorithm samples perturbations of the object shape over the uncertainty distribution (see Figure 5), and simulates a set of candidate grasps (generated for the nominal object) on the perturbations to determine their performance under the uncertainty. A grasp quality measure, defined as a lower bound on the probability of grasp success, is calculated as a weighted percentage of predicted successes over the total number of perturbations tested, where the successes and failures are weighted by the probability of that perturbation occurring.

We generalize the sampling-based approach to 3D parts with pose uncertainty as follows. We sample over perturbations in object pose and test our candidate grasps over these perturbations. The analysis performed by GraspIt! returns a quality score based on contacts in wrench space [26] for each grasp and perturbation, and the weighted average of quality score for a grasp over all perturbations (where the weights are as above, the probability of a perturbation occurring) is used as the quality measure for each candidate grasp.

C. Online Phase: Pose Estimation and Grasp Selection

If the cloud identifies the object and returns its reference data, The robot initiates a grasp by first estimating the pose of the object. This is performed with a least-squares fit between the measured 3D point cloud and the reference point set using the iterative closest point method (ICP) [32].

We use the ICP implementation from PCL. The ICP algorithm requires an initial estimate to reliably converge, so we run ICP with the reference point set over a series of 300 upright and horizontal poses. Then, the initial estimate is computed by aligning the reference point set to the detected point cloud such that the reference point set is on the work surface and the sides of the point cloud and point set are roughly aligned. The ICP algorithm generates a confidence score and the best fit is chosen.

Using the estimated object pose, a candidate grasp is chosen from the candidate grasp set based on feasibility as determined by the grasp planner. The grasp is planned using the inverse kinematics planner from OpenRAVE, a robotics motion-planning library [4]. Once the grasp is executed, the outcome data: the image, object label, detected point cloud, estimated pose, selected grasp, and success or failure of the grasp is uploaded to the key-value store server for future reference.

V. Experiments

Fig. 6. The set of six objects used for testing. The objects were selected as representative of common household objects and are easily graspable by a parallel-jaw gripper.

We experimented with the set of six household objects shown in Figure 6. We used the Willow Garage PR2, a two-armed mobile manipulator. We selected these objects because they represent common object shapes and are graspable by the PR2's parallel-jaw gripper. The experimental hardware setup is shown in Figure 1. We used a robot-head-mounted ASUS Xtion PRO sensor, similar to a Microsoft Kinect, as our 3D sensor, and used the PR2's built-in high-definition Prosilica camera.

A. Object Recognition Results

We evaluated the performance of the Google object recognition server using a variety of training image sets.

1) Training Images: We used the PR2's camera to capture 615 object images such as those shown in Figure 7. We took images of objects in different poses against solid black and wood grain backgrounds, and under ambient florescent lighting and bright, diffuse incandescent light.

2) Test Results: We created 4 different training sets—a set of images randomly sampled from our pool (R), and three rounds of hand-selected training images (A,B,C). We trained the server on each set and used the remaining images in our pool to evaluate recognition performance. The hand-selected sets used human intuition about what would make a representative set of images.

Table I shows the recall on the test set for the three training sets. We were able to achieve higher recall than random sampling through multiple rounds of hand-selected

Fig. 7. Example training images of the six objects.

Training Set	Size	Recall	Recall Rate	Training Time (s)	Recall Time (s)
R	228	307/387	0.79	0.45	0.29
A	92	247/422	0.59	0.40	0.29
B	52	215/422	0.51	0.39	0.28
A+B	144	317/422	0.75	0.40	0.29
C	49	199/422	0.47	0.39	0.30
A+B+C	193	353/422	0.84	0.40	0.29

TABLE I

IMAGE RECOGNITION PERFORMANCE FOR IMAGE TRAINING SETS. SET R WAS RANDOMLY SAMPLED. SETS A, B, AND C WERE HAND-SELECTED. THE AVERAGE CALL TIMES FOR TRAINING AND MATCHING A SINGLE IMAGE ARE GIVEN.

training images, but we were surprised to see that random sampling performed nearly as well (79% vs. 84%). Although there were many images for which the system was unable to make any identification, there were no false identifications among the images we tested. For images where no object was recognized, such as those shown in Figure 8, lighting or the camera angle often obscured the text on labels.

Fig. 8. Example images where no object could be identified.

B. Pose Estimation Results

We evaluated the system's pose estimation using 15 stable poses for each object. We manually determine failure when the estimated pose is more than 5 mm or 5 degrees from the true pose. The soap box gave the best results due to it's planar surfaces and well-defined edges. We observed that rotational symmetries of the object can cause the ICP algorithm to find a well-fitting but incorrect pose; most often this occurred with the estimated pose being inverted vertically from the true pose. For example, the shape of the mustard bottle is roughly symmetric above and below the waist of the bottle if the spout is disregarded. The ICP algorithm discards the spout this as part of its outlier rejection step, and produces a high quality score with an inverted pose for this object.

Object	Total Trials	Failures	Failure Rate	Average Time (s)
Air freshener	15	2	0.13	7.4
Candy	15	0	0.00	1.4
Juice	15	1	0.07	10.2
Mustard	15	2	0.13	10.6
Peanut butter	15	2	0.13	2.1
Soap	15	0	0.00	3.6

TABLE II

POSE ESTIMATION RESULTS. WE MANUALLY DETERMINE FAILURE WHEN THE ESTIMATED POSE IS MORE THAN 5 MM OR 5 DEGREES FROM THE TRUE POSE.

C. Grasp Results

Object	Candidate Grasp Set Size	Total Trials	Failures	Failure Rate
Air freshener	76	13	2	0.15
Candy	30	15	3	0.20
Juice	105	14	1	0.07
Mustard	61	13	3	0.23
Peanut butter	80	13	2	0.15
Soap	30	15	0	0.00

TABLE III

GRASP EXECUTION RESULTS. FOR CASES WHERE POSE ESTIMATION IS SUCCESSFUL, THE SYSTEM EXECUTES A GRASP AND ATTEMPTS TO LIFT THE OBJECT OFF THE WORKSURFACE. WE DECLARE FAILURE IF THE ROBOT DOES NOT ACHIEVE A GRASP OR DROPS THE OBJECT DURING OR AFTER LIFTING.

We evaluated grasping with cases where pose estimation is successful by having the system execute a grasp and attempt to lift the object off the worksurface. We declare failure if the robot does not achieve a grasp or drops the object during or after lifting. For some objects such as the air freshener and mustard bottle, small errors in pose estimation had a significant effect on grasp outcome. This is not surprising since in stable horizontal poses, the mustard bottle is nearly the width of the PR2's gripper opening. For the air freshener, the rounded and curved shape made it prone to rolling out of the gripper as it closed.

VI. DISCUSSION AND FUTURE WORK

This paper presents a system architecture, implemented prototype, and initial experiments for Cloud-based grasping.

Object recognition is performed off-board the robot using a variant of the Google object recognition engine which we treated as a black-box. We incorporated existing tools for pose estimation and grasping and introduce a sampling-based approach to 3D grasping. While we are encouraged by the initial results, much remains to be done to allow such a system to be scaled up and used with many robots and humans that contribute to shared computation and data analysis.

Our next step is to do more experiments with larger object sets and study how more accurate CAD models, which could be downloaded from commercial databases, may affect pose estimation and grasping. We will also refine each phase of the algorithm to incorporate confidence values which we did not use in this version. For example when the image recognition system returns results with low confidence, the robot might adjust the lighting or move the object or its own camera to obtain subsequent images, and similar feedback loops can be developed for pose estimation and grasping.

VII. Acknowledgments

We thank Dmitry Berenson for valuable discussions, Zoe McCarthy for technical assistance, and Pieter Abbeel for providing access to the PR2. This work was supported in part by NSF Award 0905344.

References

[1] Google Cloud Storage. http://cloud.google.com/products/cloud-storage.html.

[2] Google Goggles. http://www.google.com/mobile/goggles/.

[3] IEEE Society of Robotics and Automation Technical Committee on Networked Robotics. http://tab.ieee-ras.org/committeeinfo.php?tcid=15.

[4] OpenRAVE. http://openrave.org/.

[5] PCL: The Point Cloud Library. http://pointclouds.org.

[6] Willow Garage PR2. http://www.willowgarage.com/pages/pr2/overview.

[7] Rajesh Arumugam, V.R. Enti, Liu Bingbing, Wu Xiaojun, Krishnamoorthy Baskaran, F.F. Kong, A.S. Kumar, K.D. Meng, and G.W. Kit. DAvinCi: A Cloud Computing Framework for Service Robots. In *IEEE International Conference on Robotics and Automation*, pages 3084–3089. IEEE, 2010.

[8] A. Bicchi and V. Kumar. Robotic grasping and contact: a review. In *IEEE International Conference on Robotics and Automation*, pages 348–353. IEEE, 2000.

[9] Matei Ciocarlie, Kaijen Hsiao, E. G. Jones, Sachin Chitta, R.B. Rusu, and I.A. Sucan. Towards Reliable Grasping and Manipulation in Household Environments. In *Intl. Symposium on Experimental Robotics*, pages 1–12, New Delhi, India, 2010.

[10] Matei Ciocarlie, Caroline Pantofaru, Kaijen Hsiao, Gary Bradski, Peter Brook, and Ethan Dreyfuss. A Side of Data With My Robot. *IEEE Robotics & Automation Magazine*, 18(2):44–57, June 2011.

[11] Alvaro Collet, Dmitry Berenson, Siddhartha S. Srinivasa, and Dave Ferguson. Object Recognition and Full Pose Registration from a Single Image for Robotic Manipulation. In *IEEE International Conference on Robotics and Automation*, pages 48–55. IEEE, May 2009.

[12] P Duygulu, K Barnard, J F G De Freitas, and D A Forsyth. Object Recognition as Machine Translation: Learning a Lexicon for a Fixed Image. In *European Conference on Computer Vision*, pages 97–112, 2002.

[13] Roy T. Fielding and Richard N. Taylor. Principled Design of the Modern Web Architecture. *ACM Transactions on Internet Technology*, 2(2):115–150, May 2002.

[14] Stephan Gammeter, Alexander Gassmann, Lukas Bossard, Till Quack, and Luc Van Gool. Server-side Object Recognition and Client-side Object Tracking for Mobile Augmented Reality. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, number C, pages 1–8. IEEE, June 2010.

[15] Jared Glover, Daniela Rus, and Nicholas Roy. Probabilistic Models of Object Geometry for Grasp Planning. In *Robotics: Science and Systems*, Zurich, Switzerland, 2008.

[16] Corey Goldfeder and Peter K. Allen. Data-Driven Grasping. *Autonomous Robots*, 31(1):1–20, April 2011.

[17] Y. Hirano, K. Kitahama, and S. Yoshizawa. Image-based Object Recognition and Dexterous Hand/Arm Motion Planning Using RRTs for Grasping in Cluttered Scene. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2041–2046. IEEE, 2005.

[18] Katsushi Ikeuchi. Generating an interpretation tree from a CAD model for 3D-object recognition in bin-picking tasks. *International Journal of Computer Vision*, 1(2):145–165, 1987.

[19] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Hamming Embedding and Weak Geometric Consistency for Large Scale Image Search. In *European Conference on Computer Vision*, pages 304–317, Marseille, 2008.

[20] Ben Kehoe, D Berenson, and K Goldberg. Estimating Part Tolerance Bounds Based on Adaptive Cloud-Based Grasp Planning with Slip. In *IEEE International Conference on Automation Science and Engineering*. IEEE, 2012.

[21] Ben Kehoe, Dmitry Berenson, and Ken Goldberg. Toward Cloud-based Grasping with Uncertainty in Shape: Estimating Lower Bounds on Achieving Force Closure with Zero-slip Push Grasps. In *IEEE International Conference on Robotics and Automation*, pages 576–583. IEEE, May 2012.

[22] James J. Kuffner. Cloud-Enabled Robots. In *IEEE-RAS International Conference on Humanoid Robots*, Nashville, TN, 2010.

[23] K. Lai and D. Fox. Object Recognition in 3D Point Clouds Using Web Data and Domain Adaptation. *The International Journal of Robotics Research*, 29(8):1019–1037, May 2010.

[24] D.G. Lowe. Object Recognition from Local Scale-invariant Features. In *IEEE International Conference on Computer Vision*, pages 1150–1157 vol.2. IEEE, 1999.

[25] G. McKee. What is Networked Robotics? *Informatics in Control Automation and Robotics*, 15:35–45, 2008.

[26] A.T. Miller and P.K. Allen. Examples of 3D Grasp Quality Computations. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1240–1246. IEEE, 1999.

[27] A.T. Miller and P.K. Allen. GraspIt! A Versatile Simulator for Robotic Grasping. *IEEE Robotics & Automation Magazine*, 11(4):110–122, December 2004.

[28] Antonio Morales, Tamim Asfour, Pedram Azad, Steffen Knoop, and Rudiger Dillmann. Integrated Grasp Planning and Visual Object Localization For a Humanoid Robot with Five-Fingered Hands. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5663–5668. IEEE, October 2006.

[29] D Nister and H Stewenius. Scalable Recognition with a Vocabulary Tree. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2161–2168. IEEE, 2006.

[30] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object Retrieval with Large Vocabularies and Fast Spatial Matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, June 2007.

[31] Ariadna Quattoni, Michael Collins, and Trevor Darrell. Conditional Random Fields for Object Recognition. In *Advances in Neural Information Processing Systems*, pages 1097–1104, 2004.

[32] Szymon Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *International Conference on 3-D Digital Imaging and Modeling*, pages 145–152. IEEE Comput. Soc, 2001.

[33] 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, October 2007.

[34] A. Saxena, Justin Driemeyer, and Andrew Y. Ng. Robotic Grasping of Novel Objects using Vision. *The International Journal of Robotics Research*, 27(2):157–173, February 2008.

[35] A Sorokin, D Berenson, S S Srinivasa, and M Hebert. People helping robots helping people: Crowdsourcing for grasping novel objects. *2010*

IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 2117–2122, October 2010.

[36] Alexander Sorokin and David Forsyth. Utility Data Annotation with Amazon Mechanical Turk. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, number c, pages 1–8. IEEE, June 2008.

[37] George Stockman. Object Recognition and Localization via Pose Clustering. *Computer Vision, Graphics, and Image Processing*, 40(3):361–387, December 1987.

[38] Luis von Ahn. Human Computation. In *Design Automation Conference*, page 418, 2009.

[39] Markus Waibel. RoboEarth: A World Wide Web for Robots. `http://spectrum.ieee.org/automaton/robotics/artificial-intelligence/roboearth-a-world-wide-web-for-robots`, 2011.