Object detection methods for robot grasping: Experimental assessment and tuning

Ferran RIGUAL [a,1], Arnau RAMISA [a], Guillem ALENYA [a] and Carme TORRAS [a]
[a] *Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Barcelona*

Abstract. In this work we address the problem of object detection for the purpose of object manipulation in a service robotics scenario. Several implementations of state-of-the-art object detection methods were tested, and the one with the best performance was selected. During the evaluation, three main practical limitations of current methods were identified in relation with long-range object detection, grasping point detection and automatic learning of new objects; and practical solutions are proposed for the last two. Finally, the complete pipeline is evaluated in a real grasping experiment.

Keywords. Object detection, Grasping, Robotics

1. Introduction

Service robotics has been the center of an intense development effort for the past few years. Giving robots the ability to move around a domestic environment safely, performing mobile manipulation tasks with household objects, requires processing and making sense of a significant amount of perceptual data. One important knowledge item that has to be extracted from the stream of perceptual data is the location of the household objects that the robot needs to interact with to perform its tasks, which usually involve manipulation.

In this work we investigate current possibilities of object perception methods for mobile manipulator robots from a hands-on perspective. In such scenario, the mobile robot has to identify objects that can be far away, approach them, and perform the grasp. We identify three major problems for practical use of such implementations, namely long-range object detection, automatic object learning and grasping point selection. We provide practical solutions to the two last problems, and evaluate them in a real grasping experiment.

The scenario we are considering is a service robot, such as the REEM from PAL Robotics, in a typical household environment containing textured and untextured objects (see Figure 1 on the left). For practical reasons, we evaluate the proposed methods with our lab manipulator, which is a static WAM arm with a Barret hand and a Kinect camera

[1] Corresponding Author: Ferran Rigual, Institut de Robòtica i Informàtica Industrial, CSIC-UPC. Parc Tecnològic de Barcelona. C/ Llorens i Artigas 4-6. 08028 Barcelona. E-mail: frigual@iri.upc.edu.

with a viewpoint of the scene similar to that of a service robot (see Figure 1 on the right). We use both a Kinect and an ASUS Xtion Pro in our experiments; however, not all evaluated object detection algorithms use the 3D data provided by the camera.

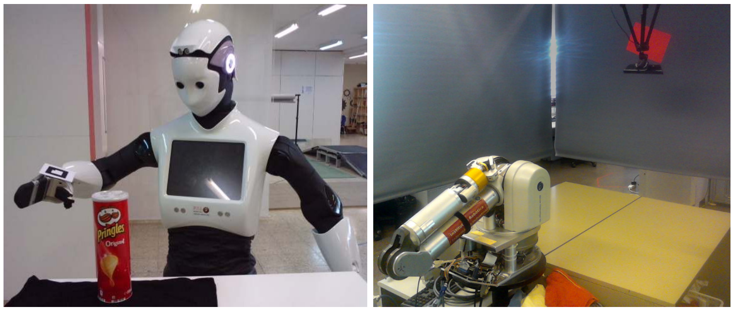


Figure 1. The REEM service robot grasping an object (left), and the manipulation scenario in the lab (right).

Perception of objects for the purpose of manipulation imposes additional restrictions: the 6 DoF (Degrees of Freedom) pose relating the detected object and its model have to be determined in order to compute grasp affordances and reliably execute the grasp action. In the past years, several competitions have been started to foster the development of this component, necessary for service robotics deployment [1,2,3]. However, it may be difficult for a method to correctly estimate the full pose of an object, but it may still give a reasonable location. In this work we experiment with a simple method to obtain a grasping point and grasp direction using 3D data and an initial location provided by the object detector.

Another critical element for these detection methods to be usable in practice is the difficulty of training for new objects. A robot should be able to learn new objects in a semi-autonomous way. Several works aimed at autonomous object model acquisition for mobile robots. Kim et al. [4] presented a method to learn new objects from scratch, by simply showing the object to the system in different poses and scales. The system used active tracking and depth data coming from a stereo system with vergence control to segment the object from the background, and was tested with a hundred different household objects, showing around 80% recognition rate on average. The drawbacks of the system were that a certain amount of texture in the objects was required, and the output of the object recognition system did not provide the 6DoF object pose, necessary for grasping applications.

The RoboEarth project [5] aims to create a “World Wide Web” for robots, where they can share their knowledge of the world, therefore reducing the burden for the robots of learning everything about their workspace. However this does not obviate having to learn the objects in the first place and, given the abundance of designs of e.g. supermarket products, and the rate at which new designs are created, it seems that a good deal of new objects will have to be learned by every robot anyway.

Finally, another problem for practical object recognition in real environments is that of “far away detection”, or how to deal with objects in the images for which no more than a few pixels are available to perform recognition, while still being crucial for the task at hand. This type of problem was the one that made impossible for any of the 2011 participants in “solutions in perception challenge” to reach the number of detected objects necessary to win the challenge. Sjö et al. [6] propose a method to recognize and localize objects in a realistic house environment. They combined an attention mechanism based on Receptive Field Co-occurrence Histograms to spot the far away objects and, later, the object was captured in a higher quality image by zooming to the appropriate region of the room, and a standard SIFT-based object detection that was used for the final recognition (SIFT stands for Scale-Invariant Feature Transform, is an algorithm in computer vision to detect and describe local features in images.).

The rest of the paper is divided in the following sections: in Section 2 we review public implementations of current object detection methods; in Section 3 the evaluated methods are compared; MOPED is selected as the best performing one and it is explained in more detail in Section 4; in Section 5 a solution for automatically training object models for MOPED is proposed and evaluated in Section 6; in Section 7 an approach to compute a reliable grasping point from object detections is proposed and tested in real grasping experiments and, finally, in Section 8, the conclusions of the work are presented.

2. Evaluated methods

We have surveyed the existing implementations of state of the art object detection methods and five candidates have been selected. Since the search scope has not been restricted, methods that work with different strategies have been found. We could roughly distinguish between two different categories:

1. When looking for objects in images, the approach of **matching feature descriptors** takes the most representative points in the image and compares them to the features of the object. The selected algorithms belonging to this approach are MOPED, RoboEarth and ORTK.
2. The **region comparison** approach focuses on one different region of the image at every step and tries to compare this whole region with the image of the object to find. GIST and Color histograms belong to this category.

Collet et al. [7] presented **MOPED**, an object recognition and pose estimation method for manipulation. In the training phase it models a 3D representation of the objects to detect. Later, the models are found in the image through feature matching and an iterative hypothesis clustering and refinement step. The authors have focused on implementing a lot of hardware-based speed optimizations (e.g. parallelization, combined use of GPU/CPU).

Ramisa et al. [10] implemented the **ORTK** method following the object recognition pipeline proposed by Lowe [11]. After keypoint matching, the Generalized Hough Transform is used to cluster matches defining a consistent transformation, and Random Sample Consensus and Iterative Reweighted Least Squares are used to create, refine and discard hypotheses of recognized objects up to an affine transformation.

The Kinect Object Detector [5] is part of the **RoboEarth** project, and its goal is to build up a huge shared knowledge base of objects for robotic systems. The database system has information about different objects, and it uses this information in order to recognize them. It also performs pose estimation so that the robot can grasp the detected objects. To train the objects, these have to be placed upon the RoboEarth pattern and be recorded with the sensor camera from multiple views.

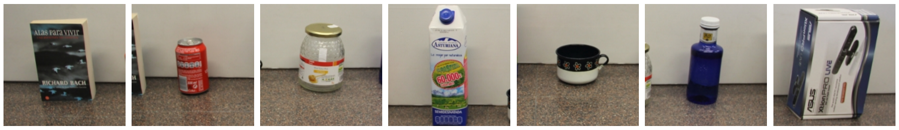


Figure 2. Objects used in the initial evaluation.

A well known technique to detect objects, specially in low-quality images, are **color histograms**. First the “color spectrum” is divided into several histogram bins, and each pixel votes for the appropriate one. In order to find object candidate location, a grid of sub-windows is defined over the test image. For every sub-window, the color histogram is computed and compared to the histogram corresponding to the training image. We have evaluated this technique using the implementation available in the OpenCV library [12].

The Global Invariant Scale Transform (**GIST**) [13] is a technique to compute a descriptor for a full image commonly used in image retrieval. Similarly to the case of color histograms, we divide the image in overlapping sub-regions, and we compare the GIST descriptor of each of the sub-regions with a reference one. The GIST descriptor is constructed from gradient and color information.

3. Initial evaluation

In this section the candidate methods are evaluated. To compare their results, we have used the same dataset on all of them, with the same conditions on the test images. We trained every method individually, and tried to achieve the best possible performance.

The dataset contains seven objects, with different characteristics regarding texture, size and color, as can be seen in Figure 2. All of them correspond to a household environment and could be used by service robots in their everyday work. There are some textureless objects which are more likely to be found by a recognizer not based on feature descriptors. Size goes up to the size of a regular box of breakfast cereals. Some colors can be found on multiple objects, and objects are of different colors as a rule. The test set consists of four images that show the images of all the objects of the dataset and some background. The images are taken from the same perspective and four distances: close view, one, two and three meters away.

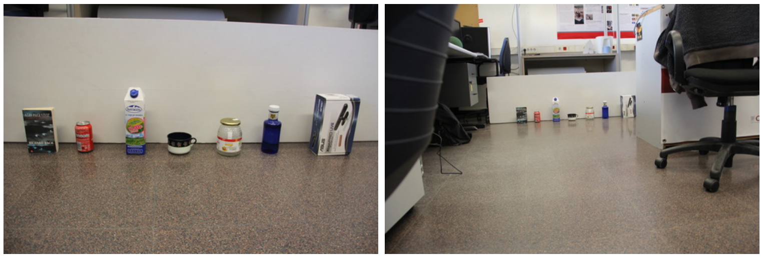
The results could be separated into two sets according to the distance: close detections, when objects are less than two meters away, and far detections when objects are at two or three meters. This way different methods could be used depending on the situation. Since there were no controversial detections, evaluation was done at hindsight. MOPED is the method obtaining best results at both distances, as can be seen in Table 1. Regarding close distance (Figure 3.a), there are several methods achieving good results: Color histograms are just one detection below MOPED (out of eight). However, MOPED gets more advantage as distance increases (as can be seen in Figure 3.b), achieving five far detections while GIST and color histograms only get two (the rest of methods can not detect anything). Given these results, we selected MOPED for the following tests.

4. MOPED

Collet et al. proposed MOPED [7], an object recognition and full 6DoF pose estimation method from a single image that uses SIFT features [11] to find correspondences between

Table 1. Initial evaluation results: Number of correct detections for each distance (out of seven), using the initially selected methods.

	Moped	**RoboEarth**	**Color**	**Gist**	**Ortk**
Close distance	5	3	5	4	3
1m	3	2	2	2	2
2m	3	0	1	2	0
3m	2	0	1	0	0
Total	**13**	**5**	**9**	**8**	**5**



(a) (b)

Figure 3. Test images: (a) close distance (b) three meters distance.

the objects in the scene and the learned models. It is based on the method of Gordon and Lowe [14] and incorporates a model alignment step for accurate localization, automatic initialization and the combination of RANSAC with Mean Shift clustering to improve the performance of the method in the case of multiple instance recognition.

The method to detect known objects in a new image consists of the following steps:

1. SIFT features of the new image are matched with those of the model.
2. Clusters of keypoints with similar rotation and scaling are found. In contrast with the method of Gordon and Lowe, here the Orthogonal Procrustes Decomposition [15] is used instead of the Generalized Hough Transform.
3. RANSAC, modified with a Mean Shift clustering step to consider groups of neighboring points, and potentially reduce the search time, is used to find all instances of the objects in the image.
4. All instances of an object with similar *rotation* and *translation* are fused together.

4.1. Training MOPED

According to the recommendations of the author, to train an object model for MOPED the user has to provide between 40 and 60 pictures of the object of interest and, for each picture, an object/background segmentation mask. Then, SIFT descriptors are extracted from the images and filtered with the masks, and matches are established between every pair of images. In order to train the models of the physical objects MOPED takes advantage of an external software: Bundler [16]. Bundler is a structure-from-motion system for unordered image collections (for instance, holiday pictures from the Internet). It takes a set of related images, image features, and image matches as input, and produces a 3D reconstruction of camera and (sparse) scene geometry as output.

There are some objects that pose additional difficulties to Bundler. This is the case of objects that have the same logo multiple times, or repetitive patterns all along their texture. Visual features tend to be very similar in these cases, and can be easily mismatched. A possible solution to this problem consists in training a model for each side separately. This way the logo will be only once in each set of pictures and no mismatches will occur.

After Bundler is run, MOPED uses its output to generate a model composed of a set of keypoints describing the object. For each keypoint, the following information is recorded:

- 3D position in the final model
- Images where the keypoint has been seen, and its 2D positions
- Average error when trying to match this point in the different images
- Color of the point in the RGB format
- SIFT descriptor

With this information, MOPED can start detecting the object in new images.

5. Automation of the training process

Robots should be able to work with a large number of objects. Most current object detection methods require an extensive offline training step for every object in order to recognize it in new images. The training process depends on every method, but getting as much information as possible is a common requisite to create a robust object model. Collecting and manually annotating this information can be a tedious and time-consuming task, specially if a lot of objects have to be trained.

Automation of the training process for MOPED intends to speed up, or completely automate, the work of getting the training data for every object. As explained in Section 4.1, MOPED needs, for every image of the object, a segmentation mask and the SIFT descriptors. Utilities to take all the SIFT descriptors from the images are available, images can also be obtained in a fast way recording a video while showing the different parts of the object. This leaves the segmentation mask as the single most difficult step to automate.

The segmentation mask consists in a black and white image, where the pixels in white correspond to the object and the pixels in black to the background. The mask is used to discriminate which descriptors have to be included in the model (the ones from the object) and discarded (those from the background). The default method to obtain the masks is through a utility that allows clicking with the cursor on the contours of the object until a closed area is defined. For a cuboid-shaped object, this means that a human operator has to click at least six times for every training picture. Considering that every object needs, according to the author, between 40 and 60 photos, this can be an unfeasible task in practice.

In order to automate this tedious work, we used a rotatory circular wooden platform. The platform is connected to a servo motor, and can be controlled by software to rotate uniformly at different speeds. With this tool, we can record a video of the object, and use plane segmentation or background subtraction in order to get the segmentation masks automatically. In this work we have implemented both methods.

Plane segmentation consists in using the depth information provided by the ASUS Xtion camera to separate the object from the rotating platform. The surface of the rotatory disk is found by adjusting a plane to it, and points on top of the plane are considered to belong to the object. The drawback of this method is that, because of the camera spec-

Table 2. Comparison of training methods: results

Model	Points	Average Error	Num Views
Default	964	1.0496	2.1317
Plane Segmentation	497	1.3910	15.3823
Background Subtraction	664	0.4689	6.0211

ifications, the minimum distance between the camera and the object is 80 centimeters, which resulted in the objects occupying a small portion of the images and containing a low number of keypoints. A similar, but more elaborate method for plane segmentation is presented in [17].

In contrast to plane segmentation, **background subtraction** only uses image information. The requirements of these methods are a camera in a fixed position and orientation during the whole training acquisition process. The method consists in comparing images with and without the objects in the rotating platform. If the background is static (does not contain moving elements), the only differences in the images will be in pixels belonging to the object. With this technique the objects can be as close as necessary to the camera.

We have performed a detection test to compare these two new methods with the original one. The test scenarios show the object in two different conditions: close and far distance. A comparison of the characteristics of the obtained models has been performed in order to better understand the advantages and disadvantages of each training method. Results can be seen on Table 2. According to the results, *plane segmentation* seems the worst method and *default* is slightly improved by *background subtraction*. The characteristics analyzed are:

1. *Points*, how many points does the model have. In general, having a large number of points translates in higher probabilities of detecting the object.
2. *Average error*, the average distance between the descriptors used to match the points.
3. *Number of views*, how many times each point has been seen, on average. The more times a point has been seen, the easier it will be to identify it from various viewpoints.

Detection results show that the performance of *background subtraction* is similar to that of the *default* method, while *plane segmentation* works slightly worse at close range.

6. Experimental results

MOPED has been used with a well-known dataset in order to test its performance and make it easier to be compared. The chosen dataset for this purpose has been the Solutions in Perception Object Dataset [1]. It contains 15 objects of different shapes and colors, but similar size. It also includes 395 test scenes, from which we randomly selected 251, with a total of 356 object occurrences.

The training images show the objects alone and centered in the image while rotating on a turning plate, and no background segmentation mask is provided. Although any of the previously discussed techniques for background/foreground segmentation would be suitable for this scenario (objects fixed in the middle of the frame) we used an even simpler filter that discards those keypoints too far from the center of the image. After

Table 3. Results in the *Solutions in Perception* dataset. No false positives were found.

Model	Detected objects	Total objects	Recall
Automatic training	293	356	82.30%
Manual training	306	356	85.96%

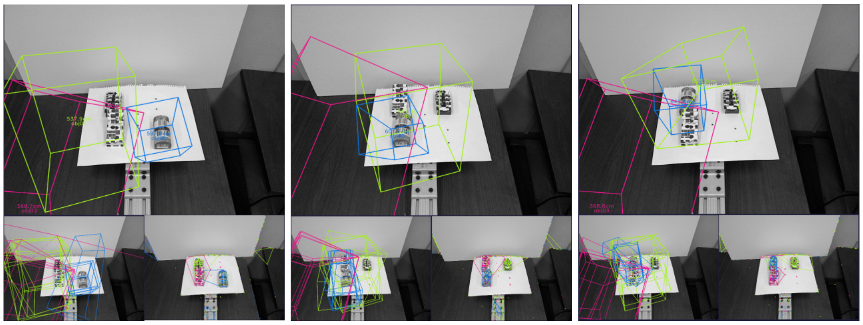


Figure 4. Example results from the Solutions in Perception Challenge using the MOPED object detector. As can be seen, objects are detected, but pose estimation is not very accurate.

using Bundler to train the object models, twelve of the fifteen objects were correctly learned. Results of this evaluation can be seen in Table 3.

7. Selecting a grasping point

Our final objective is to grasp the detected object with a robotic hand, therefore it is necessary to determine a suitable grasping point. This is usually done by computing grasp affordances on the object model [18] and then translating the computed grasp to the detected object pose. Since MOPED already estimates the 6 DoF relating the detected object and its model, we could have a precomputed set of good grasping points and select one of them depending on the current scenario. However, we have found that the object pose determined by MOPED is not entirely reliable, and an alternative solution had to be investigated. This is a difficult problem where a lot of factors intervene. In our work, we will assume that our objects are rigid and resistant, and we will consider scenarios where the object to grasp is surrounded by obstacles, but the upper part of the object will be flat and always reachable by our robot.

Given the previous conditions, the strategy selected to grasp the object consists in taking advantage of the 3D information provided by the ASUS Xtion and performing the grasp action from the upper part of the object. Considering that most objects in a household environment have a cuboid shape, the point in the middle of the upper surface is a reasonable grasping point candidate. Furthermore, it can be easily estimated as the mean of the points in the upper surface. To determine the set of points we are interested in, a plane is fit to the upper surface of the object. This way the points belonging to it can be determined, and the grasping point computed as the mean position of all of them.

Using this technique, we have performed a real grasping experiment with our manipulator robot (see Figure 5). Over the manipulator there is a fixed Kinect camera. Using the object detector, the technique described above and the depth information of the camera, the grasping point of the object has been computed and the grasp has been performed. Results of the experiments with seven different objects can be seen in Table 4. To

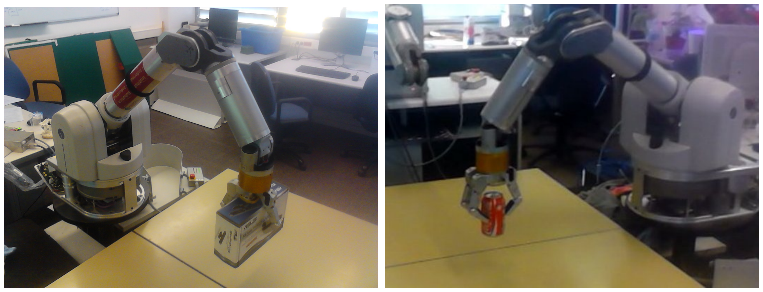


Figure 5. WAM robot grasping the ASUS Xtion box and a can of coke.

test the stability of the grasping point detection, we have computed the standard deviation between the points found for 50 consecutive frames.

Table 4. Successful grasping results and standard deviation for the detected grasping point positions of the objects in 50 consecutive frames (cm).

Model	Success/Trials	X coord.	Y coord.	Z coord.
Xtion box	5/5	1.25	2.37	1.15
Coke can	2/5	5.49	5.99	3.36
Milk	3/5	1.34	1.23	1.19
Cereal box 1	5/5	2.42	2.31	2.74
Cereal box 2	5/5	1.97	2.13	2.32
Cellphone box	5/5	2.37	3.14	3.21
Pringles can	4/5	2.11	1.19	1.07

8. Conclusions

With recent advances in navigation and mobile manipulation for service robotics, one of the most pressing current bottlenecks is object perception methods able to cope with the difficulties of unprepared house or office environments, where the robot will have to carry out tasks that involve autonomously learning novel objects, and detecting and manipulating them.

In this work we have addressed the task of setting up a practical perception system for rigid object manipulation, able to compete in current mobile manipulation challenges. For this, first a state-of-the-art object detection method has been selected among the available ones, and then it has been adapted to the requirements of our scenario by exploring methods for automatic object model acquisition and grasping point selection.

The components of the proposed method have been quantitatively evaluated in a standard robotics oriented object recognition dataset, the Solutions in Perception Challenge [1], and in various in-house datasets. Albeit modest, the practical solutions proposed in this paper identify and address some of the main, usually forgotten, limitations of current object detection methods when they are put to work.

Many future work lines follow from the contents of the paper. Testing other object detection methods, like the one of Villamizar et al. [19], or improving the automatic object model creation to work in a fully autonomous fashion (e.g. endow the robot with end-to-end object learning capabilities, and curiosity for unknown graspable entities).

Another option would be to work on *far object detection*, barely touched in this paper, but a real problem in practical situations where the perceptual workspace of the robot is limited to a few meters. Current (very expensive) workarounds to this problem involve randomly navigating the environment hoping to find, at some point, the desired objects close enough to be recognized. An alternative to investigate could be using visual attention methods to find weak object presence cues to guide this exploration.

Acknowledgements

This research is partially funded by the Spanish Ministry of Science and Innovation under projects DPI2008-06022 and MIPRCV Consolider Ingenio CSD2007-00018, and the Catalan Research Commission, and the EU Project IntellAct FP7-ICT2009-6-269959. A. Ramisa worked under the JAE-DOC grant from the CSIC and the FSE.

References

[1] "Solutions in perception challenge." `http://solutionsinperception.org/index.html`.

[2] "Mobile manipulation challenge." `http://mobilemanipulationchallenge.org/`.

[3] "Robocup@home." `www.robocupathome.org/`.

[4] H. Kim, E. Murphy-Chutorian, and J. Triesch, "Semi-autonomous learning of objects," in *Computer Vision and Pattern Recognition Workshop*, pp. 145–145, 2006.

[5] "RoboEarth kinect object detector." `http://www.ros.org/wiki/re_kinect_object_detector`.

[6] K. Sjö, D. Gálvez-López, C. Paul, P. Jensfelt, and D. Kragic, "Object search and localization for an indoor mobile robot," *Journal of Computing and Information Technology*, vol. 1, pp. 67–80, 2009.

[7] A. Collet, D. Berenson, S. S. Srinivasa, and D. Ferguson, "Object recognition and full pose registration from a single image for robotic manipulation," *International Conference on Robotics and Automation (ICRA)*, pp. 48–55, May 2009.

[8] D. Pangercic, V. Haltakov, and M. Beetz, "Fast and robust object detection in household environments using vocabulary trees with sift descriptors," in *International Conference on Intelligent Robots and Systems (IROS), Workshop on Active Semantic Perception and Object Search in the Real World*, 2011.

[9] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Conference in Computer Vision and Pattern Recognition*, vol. 2, pp. 2161–2168, 2006.

[10] A. Ramisa, D. Aldavert, S. Vasudevan, R. Toledo, and R. Lopez de Mantaras, "Evaluation of three vision based object perception methods for a mobile robot," *Journal of Intelligent & Robotic Systems*, pp. 1–24, 2012. DOI: 10.1007/s10846-012-9675-8.

[11] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[12] "The opencv library." `http://opencv.willowgarage.com/`.

[13] A. Torralba, R. Fergus, and W. T. Freeman, "80 million tiny images: a large data set for nonparametric object and scene recognition.," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 1958–70, 2008.

[14] I. Gordon and D. Lowe, "What and where: 3d object recognition with accurate pose," *Toward category-level object recognition*, J. Ponce, M. Herbert,C. Schmid and A. Zisserman (eds.), Springer-Verlag, pp. 67–82, 2006.

[15] P. Schönemann, "A generalized solution of the orthogonal procrustes problem," *Psychometrika*, vol. 31, no. 1, pp. 1–10, 1966.

[16] N. Snavely, S. Seitz, and R. Szeliski, "Photo tourism: exploring photo collections in 3d," in *Transactions on Graphics (TOG)*, vol. 25, pp. 835–846, ACM, 2006.

[17] R. Rusu, N. Blodow, Z. Marton, and M. Beetz, "Close-range scene segmentation and reconstruction of 3d point cloud maps for mobile manipulation in domestic environments," in *International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–6, 2009.

[18] A. Miller and P. K. Allen, "Graspit!: A versatile simulator for robotic grasping," *IEEE Robotics and Automation Magazine*, vol. 11, no. 4, pp. 110–122, 2004.

[19] M. Villamizar, F. Moreno-Noguer, J. Andrade-Cetto, A. Sanfeliu, "Efficient rotation invariant object detection using boosted random ferns" *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1038–1045, 2010.