Object recognition for manipulation tasks in real domestic settings: A comparative study

Luz Martínez, Patricio Loncomilla, and Javier Ruiz-del-Solar

Advanced Mining Technology Center & Dept. of Elect. Eng., Universidad de Chile
{ploncomi@ing.uchile.cl, jruizd@ing.uchile.cl}

Abstract. The recognition of objects is a relevant ability in service robotics, especially in manipulation tasks. There are many different approaches to object recognition, but they have not been properly analyzed and compared by considering real conditions of manipulation tasks in domestic setups. The main goal of this paper is to analyze some popular object recognition methods and to compare their performance in realistic manipulation setups. Object recognition methods based on SIFT, SURF, VFH, OUR-CVFH and color histogram descriptors are considered in this study. The results of this comparison can be of interest for researchers working in the development of similar systems (e.g. RoboCup @home teams).

Keywords: Object recognition, RoboCup @Home, RGB-D images, benchmark

1 Introduction

The recognition of objects is of paramount importance in service robotics, especially in manipulation tasks. The main differences with standard computer vision applications are the requirements of real-time operation with limited on-board computational resources, and the constrained observational conditions derived from the robot geometry, limited camera resolution and sensor/object relative pose. An enormous amount of articles addresses the recognition of objects in computer and robot vision. Recent approaches used in service and/or domestic robots (e.g., the ones used in RoboCup @home) are mainly based on a pipeline that first detects horizontal surfaces (e.g., a table or the floor) for restricting the search area of the possible object's positions, and then it computes features in order to recognize the objects. Popular features include the use of visual, appearance-based local interest points (keypoints) and descriptors (e.g. SIFT [7] and SURF [8]) and/or the use of 3D feature descriptors such as feature histograms obtained from range images (e.g. PFH [4] and VFH [5]). In most of the cases, the robotic platforms are equipped with RGB and/or RGB-D cameras for the data acquisition.

In this context, we believe that it is important to analyze the performance of these different approaches in domestic setups by considering real conditions. Those conditions must include variability on the typical objects to be manipulated in domestic contexts, variable illumination, dynamic backgrounds, more than one object in the robot´s field of view, search trials with more than 25 object types in the database, occlusions, typical sensors used in domestic robots (e.g. kinect sensors and low-resolution RGB cameras), and typical sensor-object pose conditions. These last two aspects are very important, because very often one of the main difficulties for the recognition task is the low resolution of the images. For instance, in Figure 2 it is shown the Bender robot as an example of a typical domestic setup. It can be observed that for the cases of objects placed on a table and on the floor, the distance between the sensors and the objects are 104cm and 177cm, respectively, and the view angle of 56° respect to the horizon. Under these conditions, and considering a typical image resolution of 640x480 or even 1280x720 pixels, the objects are observed at low resolution and in a non-frontal view.

Although in several articles the performance of object recognition methods has been analyzed and compared, these comparisons are focused on computer vision applications or they do not consider the mentioned real-world conditions, but only situations with centered high-resolution object's images.

Thus, the main goal of this paper is to analyze some popular object recognition methods and to compare their performance in realistic manipulation setups. Object recognition methods based on SIFT, SURF, VFH, OUR-CVFH and color histogram descriptors are considered in this study. The results of this comparison can be of interest for researchers working in the development of similar systems (e.g. RoboCup @home teams).

This paper is organized as follows. In Section 2 some related work is presented. In Section 3, the object recognition methods under comparison are outlined. In Section 4 the evaluation of the different methods is described. Finally, conclusions are given in Section 5.

2 Related Work

In recent years, several methodologies addressing object recognition have been developed, by both using RGB images and range images, i.e. point clouds. The development of repeatable local descriptors computed from RGB images like SIFT [7] and SURF [8] made possible the creation of a broad family of powerful methods for robust object recognition of textured objects in cluttered backgrounds. Object recognition is based on the matching between local descriptors from two different images. Coherent sets of descriptor-to-descriptor matches are found by using Hough transform clustering [7] or RANSAC [26], and they indicate possible object poses on the image. The pose of an object can be recovered by matching a test image against a set of training images captured from different viewpoints [7][1][2][3] or by reconstructing a 3D descriptor cloud by using structure from motion techniques [15][20]. However, a better recognition accuracy is obtained by generating several 2D keyframes (views) from the 3D descriptor cloud [21][22], i.e., transforming the recognition problem back from 3D to 2D, and then using 3D descriptors for retrieving an accurate pose. Also stereo images can be used for recovering the object's pose with a better accuracy [18][19]. The recognition of non-textured objects from RGB images is harder to achieve, and it requires recognizing the object's boundaries [16], although global approaches like ensemble of exemplar HoG SVMs detectors [17] that use one positive example against multiple negative ones are an interesting alternative for solving this problem.

The availability of low cost 3D capture devices like Kinect and ASUS Xtion gave rise to the development of new kind of approaches for recognizing objects; however, several of them are adaptations of 2D recognition concepts into the 3D space by using normal-based methods instead of gradient-based ones. Some methods are based on the matching of 3D local shape descriptors [23][24][25] by using variants of RANSAC [26] for pose estimation. They are able to handle clutter and occlusion, but they do not work on objects having simple shapes as they do not enable the generation of distinctive local descriptors. Also, they are sensible to both noise and variable sampling step between points. Other approaches use global or semi-global descriptors [5][6] that represent the full point cloud at once by using histograms of normals of the object, but they require to previously segment the object from the background. This is easy to enforce if the object is placed on a planar surface. An innovative approach is the use of point pair features [27] computed from pairs of points with its respective normals. The object is described globally by a set of four-dimensional training features constructed from several random pairs of oriented points, and they are stored in a two-dimensional hash table. When a test frame arrives, four-dimensional features on the point cloud are computed, and matched against the training features. Matched features vote for an object pose that is stored in a 3D accumulator, then peaks indicate possible object poses. These methods have been extended to use information from the RGB image [28][29]. This family of methods enables recognition of objects under occlusion or cluttered backgrounds.

3 Object Recognition Methods under Analysis

3.1 General Framework

The task addressed in this work is the recognition of objects for manipulation purposes. The objects are placed on a planar surface, and they are recognized by using RGB and depth images. The height of the surface (a table or the floor) can be used for segmenting zones in the image that are upper than the surface, enabling focused object recognition. There are two pipelines for object recognition: the first one is used for visual recognition, and the second one is used for point cloud based recognition.

The pipeline for visual object recognition (Fig. 1 (a)) uses a RGB image and a depth image as inputs. The depth image is used only for selecting pixels that are upper than the plane, and it is an optional stage. The RGB image is used for extracting descriptors, which are matched against descriptors stored in a training database. Extra verifications can be used for discarding unfeasible transformations. The pipeline for point cloud based object recognition (Fig. 1 (b)) uses a depth image as input. Pixels upper than the surface are selected, and a global/semi-global point cloud based descriptor is extracted and matched against descriptors stored in a training database. In this case, extra verifications are not performed.

In this work, seven visual recognition methods, two point-cloud based recognition methods and a hybrid method are compared. The visual recognition methods are: *L&R SIFT* [1][2], *L&R SIFT segm* (L&R SIFT plus object segmentation), *obj_rec_surf* [2], *obj_rec_surf segm* (obj_rec_surf plus object segmentation), *L&R*

SURF, *L&R SURF segm* (L&R SURF plus object segmentation), and color histograms. The point cloud based recognition methods are VFH [5] and OUR-CVFH [6]. The hybrid method is SIFT-VFH. These methods are described in the following sections.

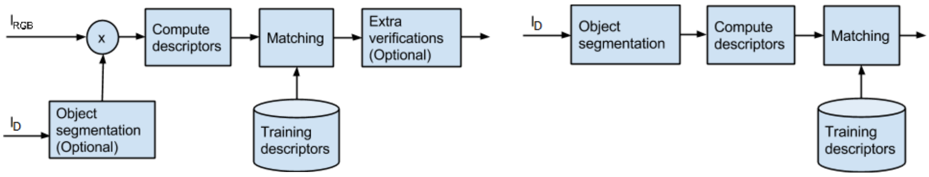

Figure 1: Block diagrams of (a) the visual object recognition pipeline, and (b) the point cloud recognition pipeline. I_{RGB}: RGB image. I_D: Depth image.

3.2 SIFT based methods

Methods based on matching local descriptors for visual object recognition are very popular. SIFT [7] is a methodology that enables computation of local descriptors by selecting interest points in the scale-space of an image, and then computing oriented, gradient-based feature vectors on image patches centered on the interest points. Descriptors from training images are stored in a kd-tree structure. Descriptors from a test image are matched against the training descriptors, and the descriptor-to-descriptor matches are clustered by using a Hough transform defined over the similarity transformation space. The *L&R SIFT* system [1] performs additional verification stages over the Hough transform bins. The stages are designed for discarding ill-defined transformations that have strong distortions, are undefined on some spatial direction or map pixels with different intensities on both images. The L&R SIFT system includes a non-maximal suppression test, a linear correlation test, a fast probability test, an affine distortion test, Lowe's probability test, a RANSAC test, a transformation fusing test, a semi-local constraints test and a pixel correlation test. Two variants of the L&R SIFT system are tested: the first variant, *L&R SIFT*, applies the object detection method over the whole image, while the second variant, *L&R SIFT segm*, applies it over a subset of the image (segmented image). The subset is obtained by detecting a planar surface in the depth image and then selecting only the regions that are upper than the plane.

3.3 SURF based methods

The SURF method [8] is similar to SIFT, but rectangular functions are used instead of Gaussians for computing the scale-space. Four variants of the SURF matching algorithm are tested: *obj_rec_surf* [2] is a methodology that matches descriptors from the test image against training descriptors by using a Hough transform based clustering, using a simple probability test for accepting or rejecting the transformations. This method was the winner of the RoboCup@Home technical challenge on 2012. It is based on the pan-o-matic [11] SURF descriptor generation implementation, which is used by Hugin [12], a panorama photo stitcher. This descriptor generator was shown to outperform other open source SURF implementations as shown by the team homer@UniKoblenz [13], and become parallelized (parallel_surf [30]) and integrated on their object recognition pipeline *obj_rec_surf*. The second variant, *obj_rec_surf segm*, is similar to *obj_rec_surf*, but SURF descriptors are computed by using the segmented image. The third variant, *L&R SURF*, is similar to *L&R SIFT*, but instead of using SIFT it uses SURF, specifically OpenSURF [14]. The fourth variant, *L&R SURF segm*, is similar *to L&R SURF*, but the descriptors are computed by using the segmented image.

3.4 VFH method

The VFH method [5] computes a global descriptor, which is formed by a histogram of the normal components of the object's surface. The histogram captures the shape of the object, and the viewpoint from which the point cloud is taken. In the first place, the angles α, ϕ and θ are computed for each point based on its normal and the normal of the point cloud's centroid c_i. The viewpoint-dependent component of the descriptor is a histogram of the angles between the vector p_c - p_v and each normal point. The other component is a SPFH estimated for the centroid of the point cloud, and an additional histogram of the distances of the points in the cloud to the cloud's centroid. The VFH descriptor is a compound histogram representing four different angular distributions of surface normals. In this work, the PCL implementation [9] is used, where each of those four histograms have 45 bins and the viewpoint-dependent component has 128 bins, totaling 308 bins.

3.5 OUR-CVFH method

OUR-CVFH method [6] computes a semi-global descriptor, and it is based on semi-global unique reference frames (SGURFs) [6] and the CVFH [10] descriptors. It exploits the orientation provided by the SGURFs to efficiently encode the geometrical properties of the object's surface. Given a surface S, the method computes the first three components of CVFH and the viewpoint component as presented in [10]. The viewpoint component is however encoded using 64 bins instead of the original 128, as normals are always pointing towards the sensor position. For each cluster c_i, a reference frame RF_i is created, then a transformation is applied to the surface S so that the points can be easily divided into the eight octants defined by the signed axes (x− , y − , z −) ... (x+ , y − , z −) ... (x+ , y + , z +). Each of the octants has an associated histogram, all of them are used for representing the surface S.

3.6 Color Histograms

Color histograms are computed by transforming the images from the RGB space into the HSV one, and then filling an histogram matrix defined over the HS space. The matrix has size 30 in the H dimension, and size 32 in the S dimension. Matching is done by using Hellinger distance, which is related to Bhattacharyya coefficient.

$$d(H_1, H_2) = \sqrt{1 - \frac{1}{\sqrt{\bar{H}_1\bar{H}_2N^2}}\sum_I \sqrt{H_1(I)H_2(I)}} \qquad (1)$$

Color histograms are computed only on segmented images, because when the full image is used the background clutter generates a large impact on the resultant histograms.

3.7 SIFT-VFH

In this hybrid method, the depth image is obtained for generating blobs, and the SIFT algorithm is applied on the image. If the blobs remain not identified, the VFH algorithm is applied on them. This algorithm is able to use both visual and shape information. The SIFT algorithm is applied before VFH because of its higher precision.

4 Evaluation

4.1 Setup and Methodology

The robot Bender from the Uchile Homebreakers team was used as platform for testing the different object recognition approaches. The robot has a Kinect camera and a RGB camera mounted over its head. Both are placed at a height of approximately 1.6[mt], pointing downwards with an angle of 56° respect to the horizon. In the reported experiments, two kinds of object placements are used: objects are placed on a table or they are placed on the floor (see Figure 2). The Kinect has a resolution of 640x480 and an angular field of view of 57° horizontally and 43° vertically. The RGB camera has a resolution of 1280 x 720 pixels, and has an angular field of view of 60° horizontally and 45° vertically. The mean distance between the robot's cameras and objects on the table is 104.1[cm], while the mean distance between the robot's cameras and objects on the floor is 176.8[cm]. As the objects are far from the camera, the area they cover on the images is small.

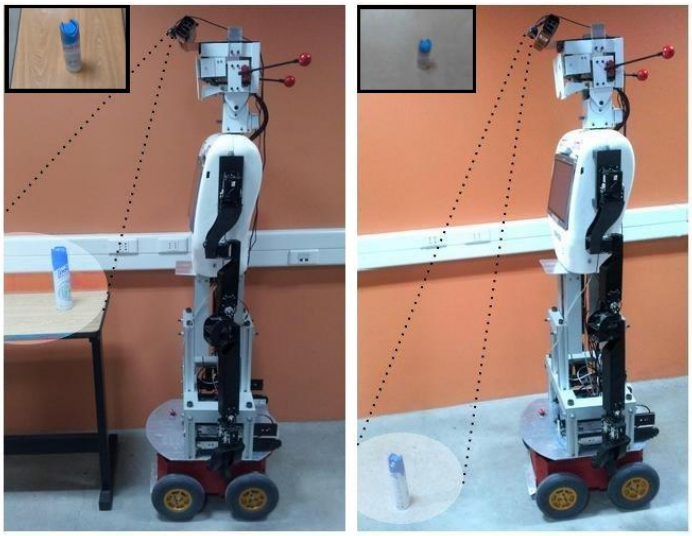

Figure 2: Bender robot observing an object on a table (see main text for details).

A set of 40 objects was selected for performing the tests; the objects are typical in a home environment (see Fig. 3). From the set of objects, 20 objects have visual textures, and the other 20 objects have uniform surfaces (no textures). For each object, 12 different views are captured by rotating the objects 30° between two consecutive frames. For each view, both an RGB and a depth image are captured. Therefore, a total of 480 RGB images and 480 depth images are used as gallery (database).

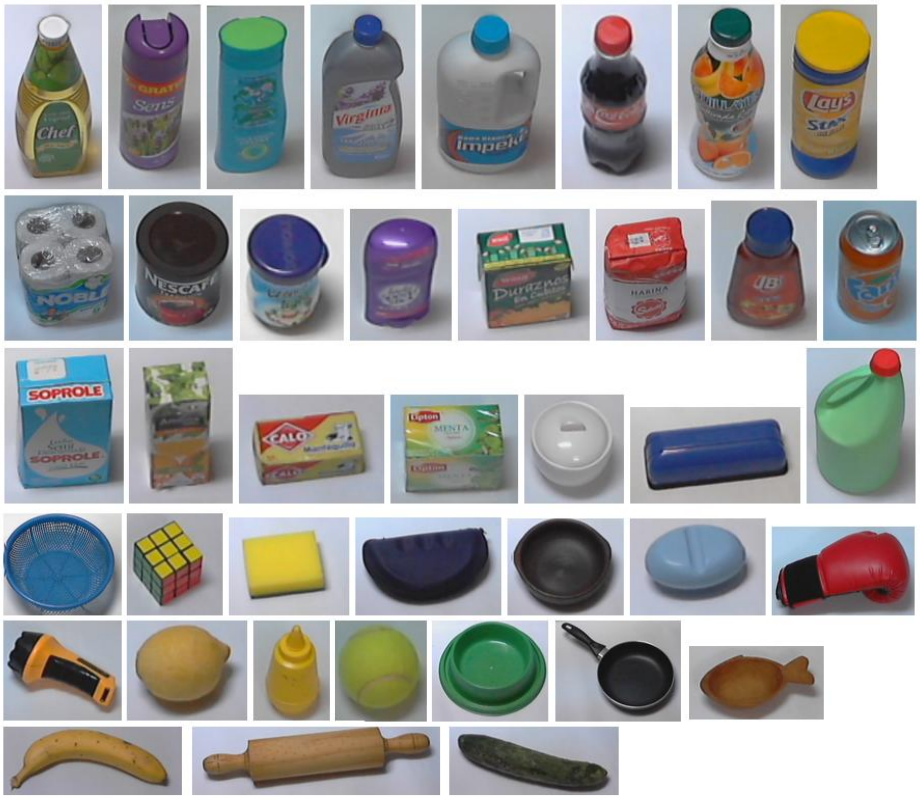

Figure 3: Set of 40 objects used for evaluating the object recognition algorithms.

Different setups are used for evaluating the performance of the different object recognition methodologies under comparison. The possible setups differ in the following conditions: (a) number of objects in each image: 1 object / 6 objects; (b) image background: white / brown / different backgrounds; (c) Illumination: normal / low; (d) occlusion: no occlusion / 50% occlusion; and (e) surface: table / floor. The selected testing setups are the following:

- S1: One object on a table, white background, normal illumination, no occlusion
- S2: One object on a table, white background, low illumination, no occlusion
- S3: One object on a table, brown background, normal illumination, no occlusion

- S4: One object on a table, different backgrounds, normal illumination, no occlusion
- S5: One object on a table, white background, normal illumination, 50% occlusion
- S6: One object on the floor, white background, normal illumination, no occlusion
- S7: One object on the floor, white background, low illumination, no occlusion
- S8: One object on the floor, brown background, normal illumination, no occlusion
- S9: One object on the floor, different backgrounds, normal illumination, no occlusion
- S10: One object on the floor, white background, normal illumination, 50% occlusion
- S11: Six objects on the table, white background, normal illumination, no occlusion
- S12: Six objects on the table, white background, low illumination, no occlusion
- S13: Six objects on the table, brown background, normal illumination, no occlusion
- S14: Six objects on the table, different backgrounds, normal illumination, no occlusion

For each of these 14 setups, 160 experiments were carried out by selecting each object 4 times; each time the object's view is chosen randomly. The random view is selected by putting the object inside the field of view of the cameras, and then selecting a random number between 0° and 360° for setting the object's orientation. The recognition is considered successful if the correct object is identitied, independently of the recovered viewpoint; i.e., the current object must be matched correctly against the 480 images (40 objects) in the database. In the case that six objects are being matched, results (success or failure) from individual matches are added. From the 160 experiments per setup, two measures, precision and recall, are used for describing the accuracy of the recognition system. When an object is put on the table or the floor, and the object recognizer is executed there are three possible outcomes: the object is successfully recognized (true positive), the object is detected but mislabeled and confused with another object (false positive) or it is not detected (false negative). From the detection statistics, precision and recall are computed as TPR/(TPR+FPR) and TPR/(TPR+FNR), respectively, with TPR the true positive rate, FPR the false positive rate and FNR the false negative rate.

4.2 Results

The experiments were carried in a ultrabook with Intel Core i7 @ 2 GHz and 2048 MB RAM using only one thread, except for the *obj_rec_surf* method that uses all of the available cores. Surface planes are detected using PCL.

Results for one object on a table. Recall/Precision results, as well as execution times are shown in Tables 1 and 2. When normal illumination conditions are considered (S1), the method *obj_rec_surf* achieves a good recall rate (0.84), but a low precision (0.72), and it works better than *obj_rec_surf segm*. The method L&R SIFT gets an acceptable recall rate (0.76) and a good precision (0.96), then it can be used as a reliable source of information for object manipulation. Under low illumination (S2) or when using variable backgrounds (S3 and S4), performance of visual detection methods falls down, and then point cloud based recognizers VFH and OUR-CVFH perform better with an acceptable recall (between 0.67 and 0.72) but with a limited precision (between 0.58 and 0.75). Under these unfavorable conditions, the best visual recognition method is L&R SIFT; it has a bad recall (between 0.22 and 0.36) but a good precision (between 0.92 and 0.97). When occlusions are present (S5), precision of point cloud based methods falls down, then the method with best recall is *obj_rec_surf seg* (0.65) but it has a bad precision (0.43) that makes it unusable for manipulation purposes. In that case (S5), the most recommendable method is L&R SIFT, it achieves a lesser recall (0.58) but a good precision (0.98). The mean of the results of recognition of one object shows that SIFT-VFH is the method with the highest recall, and it outperforms both VFH and OUR_CVFH. L&R SIFT is the best visual method by having a good recall and an excellent precision. Color histograms have a poor performance, and they are overcome by the other visual and shape based methods. OUR-CVFH is much faster than VFH. As both have similar recall and precision, the first one is recommended. *L&R SIFT segm* is much faster than L&R SIFT, but the first one has a lesser recall, and then there exists a tradeoff as no one of the methods outperforms the other. Methods *obj_rec_surf* and *obj_rec_surf seg* require a considerable processing time and have a limited precision, then the L&R SIFT variants perform better. Then, the recommended method for object recognition of objects on tables is SIFT-VFH.

Table 1: Recall/Precision for recognition of one object on a table

Setup	L&R SIFT	L&R SURF	L&R SIFT seg	L&R SURF seg	obj_ rec_ surf	obj_rec_ surf seg	VFH	OUR-CVFH	Hist.	SIFT-VFH
S1	0.76/ 0.96	0.39/ **1**	0.53/ 0.98	0.24/ **1**	0.84/ 0.72	**0.82**/ 0.51	0.63/ 0.58	0.64/ 0.65	0.45/ 0.56	0.78/ 0.91
S2	0.36/ 0.97	0.09/ **1**	0.21/ **1**	0.06/ **1**	0.29/ 0.70	0.27/ 0.41	0.67/ 0.58	**0.69**/ 0.65	0.29/ 0.53	0.66/ 0.88
S3	0.31/ 0.96	0.12/ 0.91	0.21 /**1**	0.04/ **1**	0.19/ 0.57	0.25/ 0.40	**0.72**/ 0.75	0.67/ 0.69	0.02/ 0.27	0.61/ 0.78
S4	0.22/ 0.92	0.03/ **1**	0.21/ 0.94	0.05/ **1**	0.17/ 0.75	0.29/ 0.53	**0.67**/ 0.72	**0.67**/ 0.70	0/**1**	0.66/ 0.88
S5	0.58/ 0.98	0.31/ **1**	0.44/ 0.99	0.21/ **1**	0.57/ 0.77	**0.65**/ 0.43	0.29/ 0.25	0.22/ 0.21	0.22/ 0.31	0.58/ 0.85
Mean	0.45/ 0.96	0.19/ 0.98	0.32/ 0.98	0.12/ **1**	0.42/ 0.70	0.46/ 0.46	0.6/ 0.58	0.58/ 0.58	0.20/ 0.53	**0.66**/ 0.86

Table 2: Execution time (ms) for recognition of one object on a table.

Setup	L&R SIFT	L&R SURF	L&R SIFT seg	L&R SURF seg	obj_ rec_ surf	obj_rec_ surf seg	VFH	OUR-CVFH	Hist.	SIFT-VFH
S1	1278	229	245	124	1735	2155	2180	432	**152**	464
S2	1103	325	262	**120**	1595	1422	2041	423	134	393
S3	1031	242	376	**103**	1604	1221	1882	426	115	412
S4	1253	**231**	643	525	1430	1157	3418	650	448	1485
S5	1241	372	230	203	1681	1941	2243	491	**148**	427
Mean	1181	280	351	215	1609	1579	2353	485	**199**	636

Results for one object on the floor. Recall/Precision results, as well as execution times are shown in Tables 3 and 4. As it can be observed, the point cloud based methods VFH and OUR-CVFH have a very small recall that makes them non useful for manipulation purposes, and SIFT-VFH is overcome by obj_rec_surf, then visual detection methods must be used. The methods with higher recall *are obj_rec_surf* and *obj_rec_surf segm*, but they have a limited precision. L&R SIFT has a good precision but a very low recall. Then, the analysis of the data shows that there is not a reliable way to detect objects on the floor by using a camera on the robot's head; main reason is the low resolution of the objects in the images. The use of high-resolution cameras could revert this situation.

Table 3: Recall/Precision for recognition of one object on the floor

Setup	L&R SIFT	L&R SURF	L&R SIFT seg	L&R SURF seg	obj_rec_ surf	obj_rec_ surf seg	VFH	OUR-CVFH	Hist.	SIFT-VFH
S6	0.49/ 0.97	0.06/ **1**	0.26/ **1**	0.02/ **1**	**0.77**/ 0.67	0.54/ 0.20	0.19/ 0.07	0.01/ 0.01	0.07/ 0.02	0.44/ 0.42
S7	0.19/ **1**	0.01/ **1**	0.12/ **1**	0.01/ **1**	0.47/ 0.56	**0.48**/ 0.48	0.19/ 0.09	0/ **1**	0.05/ 0.02	0.2/ 0.29
S8	0.19/ **1**	0.02/ **1**	0.08/ **1**	0.01/ **1**	0.26/ 0.52	**0.47**/ 0.28	0.15/ 0.07	0.01/ 0.01	0/ 0	0.11/ 0.23
S9	0.14/ 0.92	0.01/ **1**	0.08/ 0.93	0.01/ **1**	0.22/ 0.72	**0.45**/ 0.38	0.07/ 0.05	0.02/ 0.01	0/ 0	0.09/ 0.15
S10	0.28/ **1**	0.03/ **1**	0.17/ **1**	0.02/ **1**	0.39/ 0.57	**0.56**/ 0.22	0.07/ 0.03	0/ 0	0.08/ 0.03	0.25/ 0.37
Mean	0.26/ 0.98	0.02/ **1**	0.14/ 0.99	0.01/ **1**	0.42/ 0.61	**0.50**/ 0.31	0.13/ 0.06	0.01/ 0.01	0.04/ 0.21	0.22/ 0.28

Table 4: Execution time (ms) for recognition of one object on the floor

Setup	L&R SIFT	L&R SURF	L&R SIFT seg	L&R SURF seg	obj_rec_ surf	obj_rec_ surf seg	VFH	OUR-CVFH	Hist.	SIFT-VFH
S6	1127	270	542	**379**	1725	7469	974	617	648	2390
S7	1281	**238**	514	457	1753	5048	892	563	618	2414
S8	989	**250**	561	377	1692	5778	902	594	484	2609
S9	1125	**252**	320	275	1839	9214	725	418	441	1307
S10	992	**233**	566	440	1813	6339	1041	609	718	1619
Mean	1103	**249**	501	386	1765	6770	907	560	582	2067

Results for six objects on a table. In these experiments, six random objects on a table must be recognized. 160 tests are performed as follows: 40 times using only textured objects, 40 times using only objects without texture, and 80 times using objects with and without texture. Recall/Precision results, as well as execution times are shown in Tables 5 and 6. As it can be observed, the method with the highest recall is OUR-CVFH; however, its precision is limited (between 0.67 and 0.69). OUR-CVFH outperformes VFH. Visual methods have a poor performance in normal conditions (S11) and perform even worse in the other cases (S12, S13, S14). *L&R SIFT* outperforms *obj_rec_surf* in recall, precision and speed. *L&R SIFT* and *L&R SIFT segm* have similar accuracies, but the second one is faster. The use of segmented images in *obj_rec_surf* increases its recall in a 50%, but decreases its precision in a 40%, making it unreliable for object recognition from a single frame. It is noticeable that in general the recall of the methods decreases but their precision increments as the number of objects becomes higher. Altogether, by considering the performance in terms of precision and recall, as well as the execution time, the best method isSIFT-VFH.

Table 5: Recall/Precision for recognition of six objects on a table

Setup	L&R SIFT	L&R SURF	L&R SIFT seg	L&R SURF seg	obj_ rec_ surf	obj_ rec_ surf seg	VFH	OUR-CVFH	Hist.	SIFT-VFH
S11	0.36/ 0.97	0.18/ **1**	0.36/ 0.97	0.13/ **1**	0.36/ 0.93	0.50/ 0.52	0.47/ 0.60	0.5/ 0.69	0.25/ 0.65	**0.60**/ 0.88
S12	0.1/ 0.99	0.04/ **1**	0.08/ 0.96	0.02/ **1**	0.07/ 0.90	0.12/ 0.45	0.36/ 0.58	**0.43**/ 0.68	0.10/ 0.42	0.41/ 0.8
S13	0.15/ 0.99	0.05/ **1**	0.14/ 0.98	0.02/ **1**	0.09/ 0.85	0.15/ 0.47	0.45/ 0.63	**0.50**/ 0.69	0.04/ 0.39	0.48/ 0.8
S14	0.10/ 0.94	0.03/ **1**	0.09/ 0.99	0.01/ 0.12	0.09/ 0.96	0.15/ 0.69	0.44/ 0.63	**0.53**/ 0.69	0.01/ **1**	0.36/ 0.77
Mean	0.18/ 0.97	0.07/ **1**	0.17/ 0.97	0.04/ 0.78	0.15/ 0.91	0.23/ 0.53	0.43/ 0.61	**0.49**/ 0.69	0.10/ 0.61	0.47/ 0.81

Table 6: Execution time (ms) for recognition of six objects on a table

Setup	L&R SIFT	L&R SURF	L&R SIFT seg	L&R SURF seg	obj_ rec_ surf	obj_rec_ surf seg	VFH	OUR-CVFH	Hist.	SIFT-VFH
S11	1800	586	493	313	1851	4601	2982	912	**185**	884
S12	1385	456	553	400	1779	4964	3381	1041	**284**	1101
S13	1411	309	377	**170**	1760	4544	3035	924	185	856
S14	1839	586	996	527	1849	2077	4880	1192	**522**	2025
Mean	1608	484	605	352	1810	4047	3569	1017	**294**	1217

5 Discussion and Conclusions

Tradeoffs between recall, precision and execution time are present in the object perception task. Several different object recognition methodologies could be applied in parallel in multicore systems, and their results fused. In the case of the existence of only one core, the execution of different object recognition methodologies will slow down the object detection frame rate. The L&R SIFT detections have a high precision, then they could be used immediately for object manipulation purposes. The other algorithms give not enough precision for being used independently, then several instances of them must be run in a serial or parallel way, and then a global decision using the multiple detection results must be done. L&R SIFT outperforms obj_rec_surf in precision, accuracy and speed in all of the cases, except when the object is very far. The addition of the L&R verifications to the *obj_rec_surf* pipeline could create a system with both very high recall and precision for both near and far objects and it could be addressed in a future work.

The L&R SURF implementation have a poor performance when compared to both L&R SIFT and *obj_rec_surf*. This is caused because the OpenSURF library [14] that is used in L&R SURF generates less descriptors than the *obj_rec_surf* descriptor generator [2][11][13] when used with its default parameters. It is also reflected in the algorithm runtimes, as L&R SURF is much faster than obj_rec_surf in all of the tests. OpenSURF starts with half of the scale than obj_rec_surf, then the last generates more keypoints and is better for detecting objects far from the camera. Also, descriptors generated by *obj_rec_surf* have a better repeatability [13].

Respect to execution time, the three fastest methods are L&R SURF, L&R SURF segm and color histograms; however, their recall and precision are poor and they cannot be considered as feasible alternatives. OUR-CVFH and SIFT-VFH outperform VFH in both accuracy and speed. L&R SIFT segm is faster than L&R SIFT, but its recall is lower. The obj_rec_surf method is slower than the L&R variants, and obj_rec_surf segm is even slower.

Altogether, L&R SIFT seems to be the best visual method and SIFT-VFH performs better than the only shape-based methods by considering all time, recall and precision. L&R SIFT obtains the best performance among all SIFT and SURF methods. It is also interesting to note that the factor affecting most the performance of the methods is the low resolution of the objects in the images. Important improvement in the results of the recognition process could be obtained by using cameras having higher resolutions.

Acknowledgments

This work was partially funded by FONDECYT grant 1130153.

References

[1] Ruiz-del-Solar, J., Loncomilla, P., Devia, C.: Fingerprint Verification using Local Interest Points and Descriptors. *Lecture Notes in Computer Science* Volume 5197, 2008, pp. 519-526 (2008)

[2] Ruiz-del-Solar, J., Loncomilla, P.: Robot Head Pose Detection and Gaze Direction Determination using Local Invariant Features. *Advanced Robotics*, 23 , 2009, pp. 305-328 (2009)

[3] Seib, V., Kusenbach, M., Thierfelder, S., Paulus, D.: Object Recognition Using Hough-transform Clustering of SURF Features. RoboCup@home Technical Challenge 2012 paper (2012)

[4] Rusu R.B., Blodow, N., Marton, Z.C., Beetz, M., Aligning point cloud views using persistent feature histograms. *Proc. 21st IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS),* Nice, France, Sept. 22-26, 2008, pp. 3384-3391(2008)

[5] Rusu R.B., Bradski, G., Thibaux, R., Hsu, J.: Fast 3D recognition and pose using the Viewpoint Feature Histogram. *Proc. of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. (2010)

[6] Aldoma, A., Tombari, F., Rusu, R.B., and Vincze, M.: OUR-CVFH - Oriented, Unique and Repeatable Clustered Viewpoint Feature Histogram for Object Recognition and 6DOF Pose Estimation. DAGM/OAGM Symposium. *Lecture Notes in Computer Science Volume 7476*, pp. 113-122. Springer, (2012)

[7] Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, Volume 60 Issue 2, November 2004, pp. 91 - 110 (2004).

[8] Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: SURF: Speeded Up Robust Features. *Computer Vision and Image Understanding* (CVIU), Vol. 110, No. 3, pp. 346-359, 2008. (2008)

[9] PCL – Point Cloud Library 1.7 – http://www.pointclouds.org

[10] Aldoma, A., Blodow, N., Gossow, D., Gedikli, S., Rusu, R.B., Vincze, M., Bradski, G.: CAD-Model Recognition and 6DOF Pose Estimation Using 3D Cues. In: 3DRR Workshop, ICCV (2011)

[11] Pan-o-matic - http://aorlinsk2.free.fr/panomatic/

[12] Hugin – Panorama photo stitcher - http://hugin.sourceforge.net/

[13] Gossow, D., Paulus, D., Decker, P.: An Evaluation of Open Source SURF Implementations. *RoboCup 2010: Robot Soccer World Cup XIV* (2010)

[14] OpenSURF library - http://www.chrisevansdev.com/computer-vision-opensurf.html

[15] Zillich, M., Prankl, J., Morwald, T., Vincze. M.: Knowing your limits - self-evaluation and prediction in object recognition. *IROS 2011*, pp. 813-820 (2011)

[16] Liu, M.Y., Tuzel, O., Veeraraghavan, A., and Chellappa, R.: Fast Directional Chamfer Matching. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'10)* , Anchorage, Alaska, May 2010. (2010)

[17] Malisiewicz, T., Gupta, A., Efros, A.A.: Ensemble of Exemplar-SVMs for Object Detection and Beyond . In *ICCV 2011*. (2011)

[18] Azad, P., Asfour, T., Dillmann, R.: Stereo-based vs. Monocular 6-DoF Pose Estimation using Point Features: A Quantitative Comparison. *Autonome Mobile Systeme* (AMS). Karlsruhe, Germany. (2009)

[19] Grundmann, T., Eidenberger, R., Schneider M., Fiegert M., Wichert G.V.: Robust high precision 6D pose determination in complex environments for robotic manipulation. In *Workshop of Best Practice in 3D Perception and Modeling for Mobile Manipulation* at the *International Conference on Robotics and Automation ICRA 2010* (2010)

[20] Martinez, M., Collet, A., Srinivasa S.S.: MOPED: A Scalable and low Latency Object Recognition and Pose Estimation System. In *ICRA 2010* (2010)

[21] Kim, K., Lepetit, V., Woo, W.: Keyframe-based modeling and tracking of multiple 3D objects. *ISMAR 2010*, pp. 193-198 (2010)

[22] Kim, K., Lepetit, V., Woo, W.: Real-time interactive modeling and scalable multiple object tracking for AR. Computers & Graphics Volume 36, Issue 8, December 2012, pp. 945–954 (2012)

[23] Steder, B., Rusu, R.B., Konolige, K., Burgard, W.: Point Feature Extraction on 3D Range Scans Taking into Account Object Boundaries. In Proc. of IEEE International Conference on Robotics and Automation (ICRA 2011). (2011)

[24] Johnson, A.E. and Hebert, M.: Using spin images for efficient object recognition in cluttered 3d scenes. *PAMI*, 21(5), pp. 433–449, 1999 (1999)

[25] Mian, A., Bennamoun, M., Owens, R.: On the repeatability and quality of keypoints for local feature-based 3D object retrieval from cluttered scenes. *IJCV* (2010)

[26] Fischler, M.A., Bolles, R.C.: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Comm. of the ACM* 24 (6), pp. 381–395 (1981)

[27] Drost, B., Ulrich, M., Navab, N., Ilic, S.: Model Globally, Match Locally: Efficient and Robust 3D Object Recognition. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (CVPR) (2010)

[28] Drost, B., Ilic, S.: 3D Object Detection and Localization Using Multimodal Point Pair Features. *Second Joint 3DIM/3DPVT Conference: 3D Imaging, Modeling, Processing, Visualization & Transmission* (3DIMPVT), Zurich, Switzerland (2012)

[29] Choi, C., Christensen, H.I.: 3D pose estimation of daily objects using an RGB-D camera. *IROS 2012*, pp. 3342-3349. IEEE, (2012)

[30] Parallel SURF - http://sourceforge.net/apps/mediawiki/parallelsurf